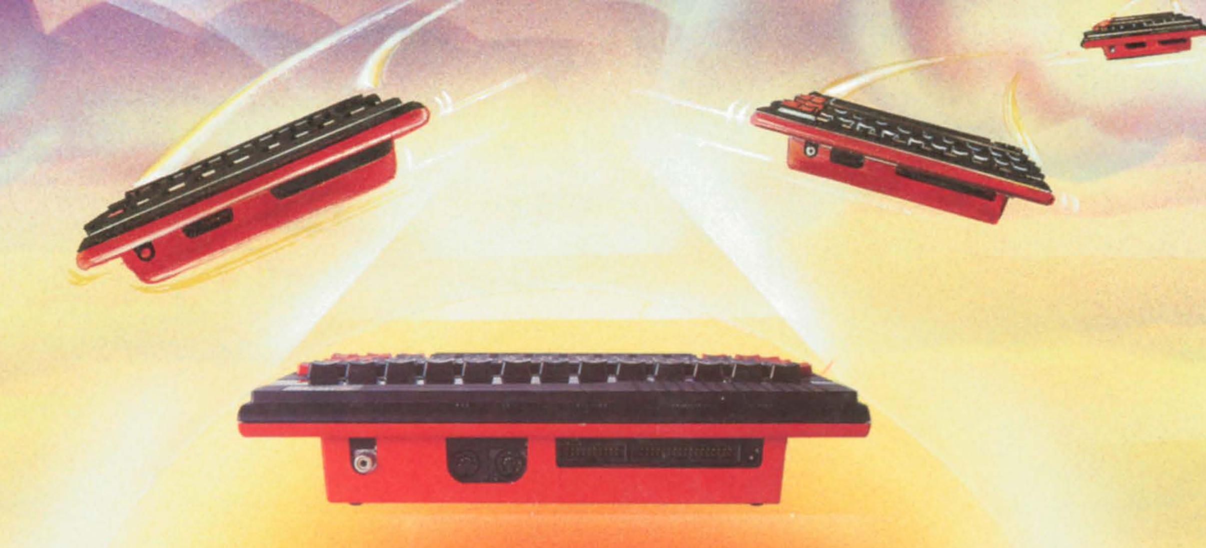
MICR'ORIC

LE MAGAZINE DES UTILISATEURS D'ORIC





N° 5 — ETE 84 — 30 F

Augmentez la capacité de vos disques

Hard copy Hires

Dollar Man Fort Oric

nt, allez-y !



La voici, votre mémoire de masse.

L'ORIC MICRO-DISC, il utilise les nouvelles disquettes de 3 pouces double face-double densité, sous carter de sécurité rigide. Capacité de 160K octets par face. Vitesse de débit 250Ko/s. Ces lecteurs sont extensibles jusqu'à 4 unités en batterie, véritable mémoire de masse pour toutes vos données et tous vos programmes.

C'est cela que vous attendiez !... alors, allez-y, maintenant !

Dans le fond, vous avez eu raison d'attendre.

Maintenant vous pouvez faire le choix définitif. Voyez : mieux qu'un ordinateur personnel, ORIC vous offre tout un système de hautes performances.

Puissant pour vous emmener de l'initiation au BASIC jusqu'à la création de progiciels de gestion (sans oublier tous les jeux !).

Fiable, ergonomique et élégant pour représenter l'informatique personnelle parvenue à sa meilleure maturité.

Accessible pour tous les budgets ; ce système ORIC ATMOS, c'est la façon de dire : "Bon, voilà ce qu'il faut pour aller de l'avant, en avoir pour son argent, et être tranquille longtemps... donc, allons-y maintenant.

IMPORTE ET DISTRIBUE PAR : ORIC-FRANCE
Z.I. «La Haie Griselle» B.P. 48 - Télex: 204 996
94470 BOISSY-ST-LEGER
Région Sud : 20, rue Vitalis 13005 MARSEILLE

MILLE ET UNE RAISONS DE SOUSCRIRE UN ABONNEMENT A MICR'ORIC

ORIC est le plus puissant des microordinateurs bon marché. MICR'ORIC est le meilleur magazine entièrement consacré à ORIC, sa technique, ses périphériques, sa programmation BASIC ou langage machine.

Numéro après numéro, MICR'ORIC vous conduira à une maîtrise approfondie de votre ORIC. Toujours bien informé, fourmillant d'idées originales grâce à la collaboration enthousiaste de nombre d'entre vous, il vous offre un recueil d'idées et de programmes très variés.

L'abonnement est fixé à ***100 F*** *pour 4 numéros ou* ***150 F*** *pour 6 numéros.*

Rythme de publication prévu : 6 numéros par an.

Pour toute correspondance indiquer votre numéro d'abonné, en particulier pour un changement d'adresse (gratuit).



BULLETIN D'ABONNEMENT

☐ *Je m'abonne aux 4 prochains numéros de MICR'ORIC* .... **100 F**
☐ *Je m'abonne aux 6 prochains numéros de MICR'ORIC* .... **150 F**
(préciser ici à partir du n° ________)

Pour compléter ma collection, je désire recevoir
les numéros suivants : ____________ *à* **35 F pièce** *port compris, soit :* ____________
(n° 2 épuisé)

Ci-joint un chèque total de : ____________
à l'ordre de ***MICR'ORIC, Z.I. La Haie Griselle, B.P. 48, 94470 Boissy-Saint-Léger***

NOM : ____________ *Prénom :* ____________

Adresse : ____________

Ville : ____________ *Code postal :* |_|_|_|_|_|

Date : ____________ *Signature :*
(des parents pour les mineurs)



LE MAGAZINE DES UTILISATEURS D'ORIC

SOMMAIRE

N° 5

MICR'ORIC est une publication d'ORIC FRANCE, département de la société A.S.N. Diffusion

Directeur :
Denis TAIEB

Rédacteur en chef :
Lucien AUGUSTONI

Ont collaboré à ce numéro :
Pierre LEDAIN
Fabrice BROCHE
Denis SEBBAG
Sylvain BRISSET
Thierry TOSELLO
Georges BARRET WAUQUIER
Christophe ANDRÉANI
Gérald AUGUSTONI
Christophe ROUX

Quelques emprunts à la revue anglaise ORIC OWNER de Tansoft Ltd.

Adresse :
MICR'ORIC
Z.I. La Haie Griselle
B.P. 48
94470 Boissy-St-Léger

Dessins : Gilles TOCUT

1re couverture, dessins, créations et conception :
STUDIO MELUN-IMPRESSIONS

Imprimerie :
MELUN-IMPRESSIONS,
18-19, rue E.-Briais, 77000 Melun
Tél. : (6) 452.04.31

Tirage : 15 000 exemplaires

Editorial



LES COLONNES D'ORIC

La revue MICR'ORIC se porte bien. Le n° 4 a donné un ton nouveau, ce n° 5 continue dans cette voie. Vous êtes de plus en plus nombreux à fournir la matière de nos colonnes. Nous publions les articles après quelques vérifications, mais nous laissons largement les auteurs responsables. Si vous trouvez des erreurs, des imprécisions, entrez dans le jeu : écrivez à MICR'ORIC. La mise en commun des trouvailles des plus communicatifs d'entre vous donne le résultat que vous pouvez constater. Ainsi nous comprenons de mieux en mieux le fonctionnement des appareils de la gamme ORIC.

Au-delà de nos colonnes, vous avez pu voir, en librairie ou chez vos revendeurs des livres variés traitant des mêmes sujets : leur nombre montre l'intérêt suscité par ORIC-1 et ATMOS. Emmanuel FLEYSSELLES qui a collaboré à MICR'ORIC a signé chez P.S.I. le mémento "Clefs pour l'Oric" (ORIC-1 et ATMOS). On y trouve des renseignements variés regroupés très clairement. De même Fabrice BROCHE (une coquille nous l'a fait appeler ROCHE dans le n° 4) qui a beaucoup d'idées prépare un livre car MICR'ORIC ne peut pas tout contenir.

La toute nouvelle société MICROPROGRAMMES 5, 82-84, boulevard des Batignolles, 75017 Paris. Tél. : (1) 293.24.58 a déjà commencé à publier des cassettes. Elle publiera des livres et des microdisques. Les auteurs peuvent soumettre leurs projets. Vous trouverez la liste des premiers titres dans ce numéro. Le programme de traitement de texte "AUTEUR" présente des caractéristiques très intéressantes, c'est une version utilisable avec un magnétocassette. Il serait agréable pour les possesseurs d'un lecteur de microdisques de disposer de ce logiciel sur disque, autorisant la manipulation plus rapide de textes encore plus longs. Rappelons que la vitesse de transfert microdisque-ORIC est de 250 000 bauds (et non 250 Ko/s), ce qui fait que la mémoire est remplie en moins de 10 secondes compte-tenu des contraintes d'accès.

Dans ce numéro vous trouverez des articles concernant le système d'exploitation de disquettes (D.O.S.) avec possibilité d'augmenter la capacité des disques.

Nous prévoyons une série d'articles permettant la copie de l'écran HIRES sur divers types d'imprimantes, en langage machine. Vous souhaitez sûrement des idées pour la gestion. Nous vous fournirons un programme clair et des méthodes pour mettre au point vos propres programmes.

MICR'ORIC



LES VARIABLES

par Pierre LEDAIN

Différents types de variables peuvent être "manipulées" par ORIC :

— les nombres réels (à virgule flottante)
— les entiers
— les chaînes de caractères

De plus, ces variables peuvent faire l'objet de tableaux, c'est-à-dire être "dimensionnées".

Il est intéressant, et même indispensable, pour programmer en langage machine, de connaître comment ces variables sont stockées et codées en mémoire.

Lorsque vous entrez en programme BASIC, il se place à partir de l'adresse # 5ØØ (soit 128Ø), ceci est bien visible sur la planche du manuel donnant le schéma de l'organisation des mémoires.

Lorsque le programme est exécuté, les variables sont placées **à la suite du programme.**

Nous trouvons d'abord les variables non dimensionnées, puis les variables dimensionnées.

- L'adresse de la 1re variable non dimensionnée est mémorisée dans les octets # 9C et # 9D soit : DEEK (# 9C) = adresse de la 1re variable non dimensionnée.
- L'adresse de la 1re variable dimensionnée (ou du 1er tableau) est mémorisée dans les octets # 9E et 9F soit : DEEK (# 9E) = adresse de la 1re variable dimensionnée.
- L'adresse de la fin des variables est mémorisée dans les octets # AØ et # A1 soit : DEEK (# AØ) = adresse de la fin des variables.

Un tout petit programme très simple permet de visionner le contenu des mémoires et pourra être utile pour illustrer les explications qui suivent.

```
1Ø HIMEM # 97FF
2Ø INPUT"A";A:INPUT"B%";B%
25 INPUT"X$";X$
3Ø FOR I=DEEK( # 9C)TO 2ØØØ
4Ø PRINT I"M :"PEEK(I);HEX$(PEEK(I)),
5Ø IF KEY$=" "THEN GET R$
5Ø IF R$="S" THEN END
7Ø NEXT I
```

- Dans les lignes 2Ø et 25 modifiez à loisir, selon ce que vous voulez observez.
- En ligne 3Ø le départ de la boucle pourra être au début des variables dimensionnées.
- On arrête momentanément le défilement en appuyant sur la barre d'espace, puis sur "S" pour arrêter le programme, sur une autre touche pour continuer.

Voyons maintenant comment sont mémorisées les variables.

1. VARIABLES NON DIMENSIONNÉES

Ces variables représentent des nombres à virgule flottante (ex. : A = 2.37) ou des nombres entiers (ex. : B% = 25) ou des chaînes de caractères (ex. : X$ = ORIC 1).

Chaque variable utilise **7 octets** pour sa mémorisation. Les **deux** premiers octets servent à **coder le nom et le type de la variable** de la manière suivante :

	1er octet	*2e octet*
nombre à virgule flottante	code ASCII de la 1re lettre du nom.	code ASCII de la 2e lettre du nom.
nombre entier	code ASCII de la 1re lettre du nom + 128 (#8Ø).	code ASCII de la 2e lettre du nom + 128 (#8Ø).
chaîne de caractère	code ASCII de la 1re lettre du nom.	code ASCII de la 2e lettre du nom + 128 (#8Ø).

Exemples :	*1er octet*	*2e octet*
Variables réelles		
AX	65 (#41)	88 (#58)
N	78 (#4E)	Ø
Variables entières		
B%	194 (#C2)	128 (#8Ø)
T4%	212 (#D4)	18Ø (#B4)
Variables chaînes		
X$	88 (#58)	128 (#8Ø)
Y2$	89 (#59)	178 (#B2)

La variable pouvant ainsi être facilement localisée, examinons comment est codé son contenu dans les cinq octets suivants.

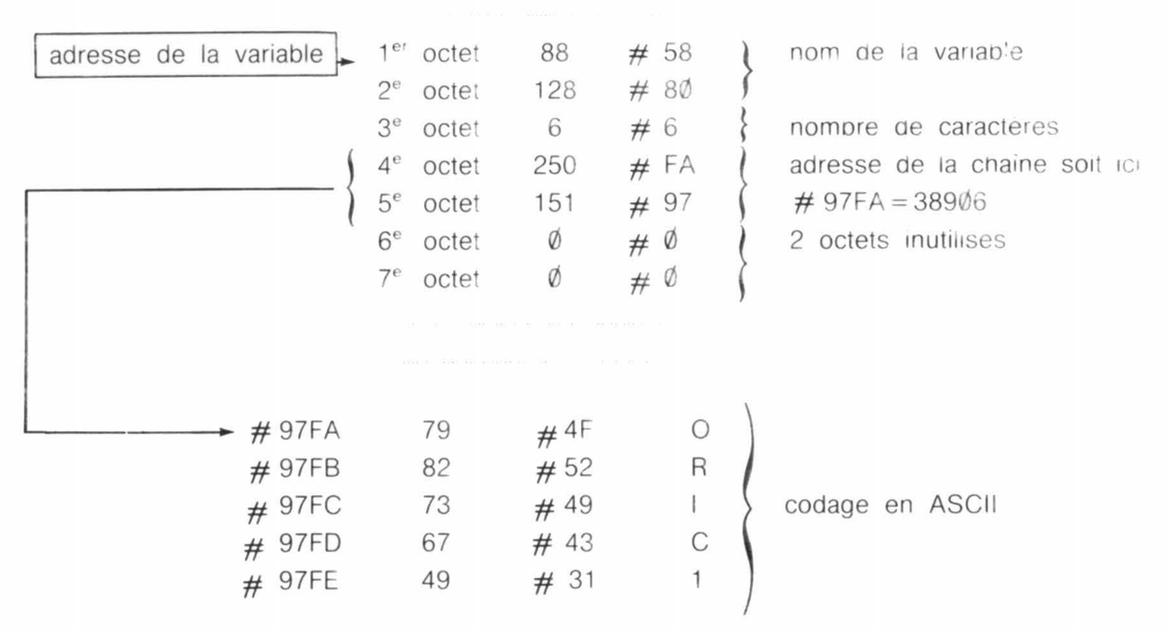
Commençons par la plus simple : la variable chaîne de caractères.



1.1. Variable chaîne de caractère

Son mode de codage est très simple et facile à comprendre avec l'exemple suivant :

X$ = "ORIC1"



adresse de la variable →

1er octet	88	# 58	nom de la variable	
2e octet	128	# 8Ø		
3e octet	6	# 6	nombre de caractères	
4e octet	250	# FA	adresse de la chaîne soit ici	
5e octet	151	# 97	# 97FA = 389Ø6	
6e octet	Ø	# Ø	2 octets inutilisés	
7e octet	Ø	# Ø		

# 97FA	79	# 4F	O	codage en ASCII
# 97FB	82	# 52	R	
# 97FC	73	# 49	I	
# 97FD	67	# 43	C	
# 97FE	49	# 31	1	

Le 3e octet donne le nombre de caractères de la variable, donc compris entre Ø et 255.

Les 4e et 5e octets donnent l'adresse à partir de laquelle on trouve la chaîne de caractères (codée en ASCII) qui est le contenu de la variable.

Le 6e et le 7e octets sont inutilisés.

Commentaire

Il faut savoir aussi que chaque nouvelle valeur de la variable X$ viendra s'empiler en mémoire (sans détruire la précédente). Toutes ces données sont empilées entre la zone qui contient le programme BASIC, à partir du plafond de la mémoire fixé par le HIMEM. Sur ORIC-1 il faut définir ce plafond au début sinon les variables envahissent la zone où sont définis les caractères créant des dessins bizarres au lieu des lettres, chiffres et signes usuels !

Il est clair que ce mode de stockage de données "chaînes" peut occuper un nombre très important de mémoires. Par exemple, dans un programme comportant plusieurs boucles qui changent le contenu d'une variable chaîne à chaque passage, les mémoires disponibles peuvent être rapidement épuisées. Dans ce cas, ORIC "fait le ménage" en éliminant les données inutiles (processus automatique analogue à celui provoqué par l'instruction FRE("")) ce qui peut demander plusieurs minutes et allonger, de ce fait, le temps d'exécution d'une manière inacceptable. Nous verrons prochainement comment contourner cette difficulté dans certain cas.



1.2. Variable entière

Son mode de codage, assez simple lui aussi, est donné ci-dessous :

1er octet	a	}	nom de la variable
2e octet	b		
3e octet	x	}	contenu
4e octet	y		
5e octet	Ø	}	
6e octet	Ø		octets inutilisés
7e octet	Ø		

Les 1er et 2e octets fournissent le nom de la variable comme indiqué précédemment.

Les 3e et 4e octets donnent le contenu de la manière suivante :

le 1er bit de x donne le signe :
si ce bit vaut Ø, le nombre est positif
si ce bit vaut 1, le nombre est négatif
il s'agit du bit de plus fort poids : 128

la valeur absolue du nombre se calcule à partir de x et y.

Ainsi : pour $Ø \leq x \leq 127$
$Ø \leq y \leq 255$

nombre positif = 256*x+y

et pour $128 \leq x \leq 255$
$Ø \leq y \leq 255$

nombre négatif = (x—256)*256+y

Exemples : B% = 281

1er octet	194	(#C2)
2e octet	128	(#80)
3e octet	1	(#1)
4e octet	25	(#19)
5e octet	Ø	
6e octet	Ø	
7e octet	Ø	

B% = 256*1 + 25 = 281

AX% = —114

1er octet	193	(#C1)
2e octet	216	(#D8)
3e octet	255	(#FF)
4e octet	142	(#8E)
5e octet	Ø	
6e octet	Ø	
7e octet	Ø	

AX% = (255—256)*256 + 142 = —114

On en déduit aisément les limites imposées par ce codage :

Le plus grand nombre positif possible est :

x = 127 } N = 127*256+255=32767
y = 255

Le plus petit nombre négatif possible :

x = 128 } n = (128—256)*+Ø = —32768
y = Ø

1.3. Variables numériques à virgule flottante

Cette variable représente donc un nombre réel quelconque, positif ou négatif. Les limites dépendent, ici aussi, du mode de codage.

Le mode de codage d'une variable réelle est un peu plus difficile à déchiffrer que ceux des autres variables.

Avec comme outil le petit programme donné plus loin et en suivant les traces de Champollion, voici ce qu'on peut découvrir :

Le codage est effectué sur 5 octets et c'est certainement la raison pour laquelle 5 octets ont été également utilisés pour les autres types de variables qui n'en demandent pas tant. Le "pas" entre chaque variable non dimensionnée étant ainsi unifié à 7 octets.

Pour comprendre l'organisation de ces 5 octets, il est nécessaire d'examiner leur contenu sous forme binaire :

1er octet	**1**	**0**	**0**	**0**	**0**	**1**	**0**	**1**	exposant
2e octet	**0**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	mantisse
3e octet	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	
4e octet	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	
5e octet	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	

ici le nombre est **15.**

Le 1er octet peut être considéré comme un exposant, les autres comme une mantisse. Pour les non-initiés le 1er octet indique la place de la virgule, les autres fournissent les chiffres.

1er octet : l'exposant

s'il est nul, le nombre est nul,
s'il n'est pas nul, sa valeur donne la position de la virgule (du point pour ORIC) du nombre considéré, ou, pour être plus concis le "poids" du 1er bit de la mantisse qui est alors égal à 2^{n-129} où n est la valeur de l'exposant.

Contrairement aux apparences, ce n'est pas très compliqué comme le montrent les quelques exemples que voici :

Exposant n	n en base 10	Valeur du 1er bit de la mantisse	Écriture du nombre en base 2
00000000	0	sans objet	0
10000000	128	$2^{128-129} = 2^{-1}$	0.1xxxx
10000001	129	$2^{129-129} = 2^{0}$	1.xxxxx
10000010	130	$2^{130-129} = 2^{1}$	1x.xxx
01111011	123	$2^{123-129} = 2^{-6}$	0.000001xxx
On en tire immédiatement les limites de ce codage			
11111111	255	$2^{255-129} = 2^{126}$	**127 chiffres** (accolade) 1xxxx.......x. ①
00000001	1	$2^{1-129} = 2^{128}$	**127 zéros** (accolade) 0.000.....01 ②

REMARQUES :

① Si la mantisse n'a que des 1, on atteint environ 2^{127} soit $1,70141183*10^{38}$.

② Si la mantisse n'a que des Ø, on atteint 2^{-128} soit $2,93873588*1Ø^{39}$.
Les emplacements xxx sont occupés par des Ø ou des 1 selon la valeur de la mantisse.

La mantisse

La mantisse est obtenue en mettant bout à bout les 2e, 3e, 4e et 5e octets, soit 32 bits. Le nombre est donc défini avec une précision de $2^{n-129-32}$

32 chiffres binaires

2 octet	3e octet	4e octet	5e octet
1			

emplacement du 1er chiffre significatif de la mantisse, au début du 2e octet. En base 2 le 1er chiffre significatif ne peut être autre que 1, en conséquence cet emplacement est utilisé par ORIC pour coder le signe du nombre.

Le chiffre Ø à cet endroit signale un nombre positif.

Le chiffre 1 indique un nombre négatif.

En utilisant le programme **"codage des nombres"*** vous pourrez très facilement vous familiariser avec les nombres en mémoire et même avec la base 2.

```
 1  REM**CODAGE DES NOMBRES**
 2 HIMEM#97FF:POKE#26A,10:PAPER0:INK2:
B=0:P=DEEK(#9C)+2:DOKE#417,P:GOSUB700
 4 POKE48000,6:A$="Modes:D=dec-bin,B=b
in-dec,M=mem,S=stop ":FORI=1TO39
 6 POKE48000+I,ASC(MID$(A$,I,1)):NEXT
 10 PRINTCHR$(131)"Mode:";:GETA$:PRINT
A$:IFA$="D"THEN90
 12 IFA$="B"THEN500
 14 IFA$="M"THEN20
 16 IFA$<>"S"THEN10ELSEPOKE48000,0:END
 18 REM**CONT.MEM.NB EN BASE 2**
 20 INPUT"Nb Dec.:";B
 30 L=0:FORI=PTOP+4:L=L+1:A=PEEK(I):PR
INTL;":¦";A;:POKE617,11
 50 PRINTHEX$(A);:POKE617,14:PRINT":";
:GOSUB100:POKE617,16:PRINTZ$;"¦"
 60 NEXT:GOSUB610:PRINTCHR$(133)"Voule
z-vous changer un octet? (O/N)"
 70 GETA$:IFA$="O"THEN74
 72 IFA$<>"N"THEN70ELSE10
 74 PRINTCHR$(135)"No de l'octet: ";:G
ETA$:A=VAL(A$):PRINTA;:INPUT"Valeur:";
K
 76 IF(A<1ORA>5ORK>255ORK<0)THENPING:G
OTO74
 78 POKEP-1+A,K:GOTO30
 90 REM**DEC.EN BINAIRE**
 95 ND=1:INPUT"Nb Dec.:";B:A=ABS(B):D=
A-INT(A):A=INT(A)
 100 Z$="":REPEAT:X=INT(A/2):Z$=MID$(S
TR$(A-2*X),2)+Z$:A=X:UNTILX=0
 145 REM**CONFORMATION A 8 BITS**
 148 IFND=1THEN300
 150 Z=LEN(Z$):IFZ<8THENFORJ=Z+1TO8:Z$
="0"+Z$:NEXT
 160 RETURN
 290 REM**"DECIMALES" BINAIRES**
 300 X=D:I=0:Z$=Z$+".":L=2^-126
 310 REPEAT:I=I+1
 320 Y=X-1/INT(2^I):IFY>=0THENX=Y:Z$=Z
$+"1"ELSEZ$=Z$+"0"
 330 UNTILX<L:PRINT"Nb Bin.:";:IFB<0TH
ENPRINT"-";
 340 PRINTZ$
 400 REM**BINAIRE EN DEC.**
 410 FORI=1TOLEN(Z$):IF(MID$(Z$,I,1)="
.")THEN420ELSENEXT
 420 N=I-2:Y=0:FORI=0TON:Y=Y+VAL(MID$(
Z$,N-I+1,1))*INT(2^I):NEXT
 430 J=N+2:K=LEN(Z$)-J:FORI=1TOK:Y=Y+V
AL(MID$(Z$,J+I,1))/INT(2^I):NEXT
 440 PRINT"Nb Dec.:";:IF(LEFT$(Z$,1)="
-"ORB<0)THENPRINT"-";
 450 PRINTY:ND=0:GOTO10
 500 REM**BIN-DEC.AVEC ENTREE**
 510 INPUT"Nb Bin.:";Z$:B=0
 520 IFLEFT$(Z$,1)="."THENZ$="0"+Z$
 530 GOTO410
 600 REM**CODAGE EN ACC1**
 610 CALL#400:POKE616,(PEEK(616)-6):PR
INT:L=-1
 620 FORI=#420TO#425:A=PEEK(I):L=L+1:P
OKE617,26:PRINTHEX$(#D0+L);": ";:GOSUB
100:PRINTZ$
 630 NEXT:PRINT"Nb Dec.:";B:RETURN
 690 REM**S/PROG.MACHINE:COD.ACC1**
 700 DATA#AD,#17,#04,#AC,#18,#04,#20,#
73,#DE,#A2,#00,#B5,#D0,#9D,#20,#04,#E8
 710 DATA#E0,#06,#D0,#F6,#60
 720 FORI=#400TO#415:READN:POKEI,N:NEX
T:RETURN
```



Après avoir lancé le programme par RUN, vous entrez le mode choisi.

D : convertit un nombre décimal (ou hexa) que vous entrez (INPUT) en binaire.

B : convertit un nombre binaire que vous entrez (INPUT) en nombre décimal.

M : vous entrez (INPUT) un nombre décimal (ou hexa) et vous obtenez son codage en mémoire ainsi que son codage effectué par ORIC pour le manipuler.

Le tableau obtenu est le suivant :

Numéro de l'octet	En base 10	En base 16	En base 2	Contenu des octets D0 à D5 en base 2*
1				# DØ
2				# D1
3				# D2
4				# D3
5				# D4
				# D5

* *après transfert du nombre en ACC1 (voir explications plus loin).*

— Vous pouvez également entrer des nombres négatifs (entiers ou non).

— Pour obtenir le nombre décimal correspondant à un nombre binaire (positif ou négatif, entier ou non), entrez bien un nombre binaire, sinon ORIC vous donnera une réponse farfelue ! Il n'y a pas de vérifications sophistiquées sur l'entrée pour ne pas alourdir le programme.

Ceci nous amène naturellement à examiner les possibilités offertes par la ROM pour manipuler des nombres.

Ce qui suit s'inspire d'un article publié par la revue ORIC OWNER de TANSOFT, directement renseignée par la firme ORIC en Angleterre.

Lorsqu'ORIC effectue des opérations sur 2 nombres, il code chacun de ceux-ci sur 6 octets. Le premier que nous appellerons ACC1 est placé de # DØ à # D5. Le deuxième que nous appellerons ACC2 est placé de # D8 à # DD.

Le mode de codage sur 6 octets s'apparente à celui exposé ci-dessus :

Par exemple on a :

DØ = 1er octet ⟶ exposant

mantisse {
- # D1 = 2e octet ⟶ sauf le 1er bit qui est ici à 1
- # D2 = 3e octet
- # D3 = 4e octet
- # D4 = 5e octet
}

D5 = 1er octet ⟶ donne le signe du nombre par son 1er bit

Nous appellerons MEM l'adresse du 1er des 5 octets codés en mémoire et déterminée par l'utilisateur et généralement entrée dans l'accumulateur (A) et le registre (Y) du 6502 (voir l'appendice K page 175 du manuel ORIC-1 ou l'annexe 8 p. 289 du manuel de l'ATMOS). Cette adresse nécessite 2 octets, nous appellerons MEM BS (octets bas) l'octet le moins significatif et MEM HT (octet haut) l'octet le plus significatif.

LISTE DES ROUTINES

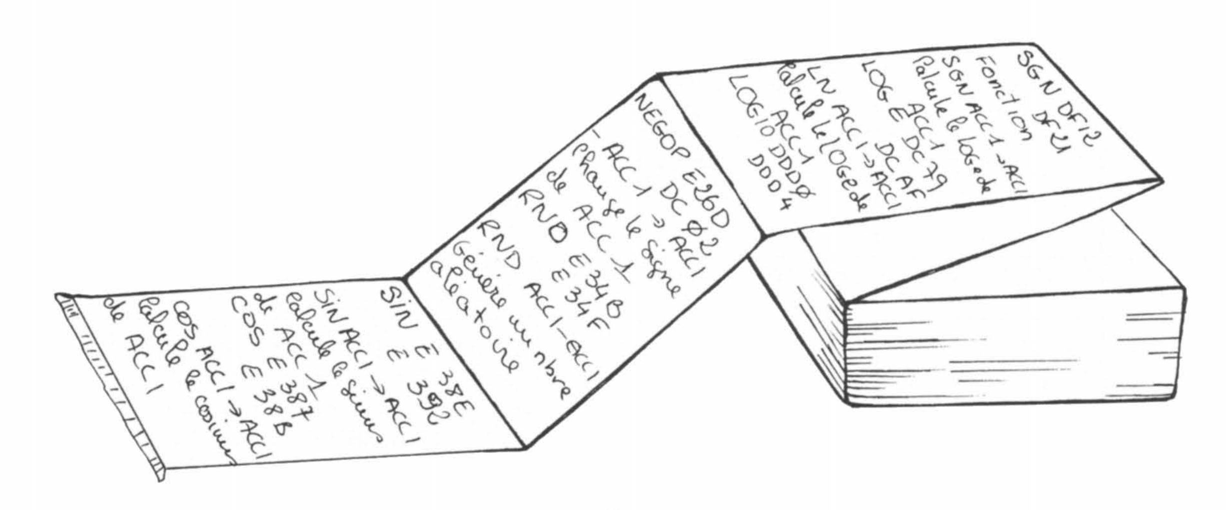
NOMS	ADRESSES		FONCTION	DESCRIPTION
	ORIC	ATMOS		
MOVFM	DE73	DE7B	MEM → ACC1	Transfère MEM dans ACC1 A = MEM BS, Y = MEM HT
CONUPK	DD4D	DD51	MEM → ACC2	Transfère MEM dans ACC2 A = MEM BS, Y = MEM HT
MOVFM	DEA5	DEAD	ACC1 → MEM	Transfère ACC1 dans MEM X = MEM BS, Y = MEM HT
MOVAF	DEDD	DEE5	ACC1 → ACC2	Transfère ACC1 dans ACC2
MOVFA	DECD	DED5	ACC2 → ACC1	Transfère ACC2 dans ACC1
FDIVT	DDE3	DDE7	ACC2/ACC1 → ACC1	Divise ACC2 par ACC1 et place le résultat dans ACC1
FDIV	DDEØ	DDE4	MEM/ACC1 → ACC1	Divise MEM par ACC1 A = MEM BS, Y = MEM HT
FDIV2	DDDA	?	ACC2/MEM → ACC1	Divise ACC2 par MEM A = MEM BS, Y = MEM HT
DIV 10	DDBF	DDC3	ACC1/10 → ACC1	Divise ACC1 par 10
SQR	E22A	E22E	SQR ACC1 → ACC1	Calcule la racine carrée de ACC1
POWER	E231	E235	ACC2 ↑ MEM → ACC1	Calcule ACC2 puissance MEM A = MEM BS, Y = MEM HT
EXP	E2A6	E2AA	EXP ACC1 → ACC1	Calcule e ↑ ACC1
INT	DFA5	DFBD	INT ACC1 → ACC1	Calcule la valeur entière de ACC1
ABS	DF31	DF49	ABS ACC1 → ACC1	Calcule la valeur absolue de ACC1
FADD	DA97	DB22	MEM+ACC1 → ACC1	Additionne MEM à ACC1 A = MEM BS, Y = MEM HT
FADDH	DA79	DBØ4	ACC1+0.5 → ACC1	Additionne 0.5 à ACC1
F SUB	DA8Ø	DBØB	MEM - ACC1 → ACC1	Soustrait ACC1 de MEM A = MEM BS, Y = MEM HT
F ADDA	DA9A	DB25	ACC2+ACC1 → ACC1	Calcule ACC2 + ACC1
F SUBA	DA83	DBØE	ACC2 - ACC1 → ACC1	Calcule ACC2 - ACC1
F MULT	DCB7	DCED	MEM*ACC1 → ACC1	Multiplie MEM par ACC1 A = MEM BS, Y = MEM HT
MUL 10	DDA3	DDA7	ACC1*10 → ACC1	Multiplie ACC1 par 10

LISTE DES ROUTINES (suite)

NOMS	ADRESSES		FONCTION	DESCRIPTION
	ORIC	ATMOS		
SGN	DF12	DF21	SGN ACC1 ⟶ ACC1	Calcule le signe de ACC1
LOG E	DC79	DCAF	LN ACC1 ⟶ ACC1	Calcule le LOGe de ACC1
LOG 10	DDDØ	DDD4	LOG ACC1 ⟶ ACC1	Calcule le LOG_{10} de ACC1
NEGOP	E26D	DCØ2	- ACC1 ⟶ ACC1	Change le signe de ACC1
RND	E34B	E34F	RND ACC1 ⟶ ACC1	Génère un nombre aléatoire
SIN	E38E	E392	SIN ACC1 ⟶ ACC1	Calcule le sinus de ACC1
COS	E387	E38B	COS ACC1 ⟶ ACC1	Calcule le cosinus de ACC1
TAN	E3D7	E3DB	TAN ACC1 ⟶ ACC1	Calcule la tangente de ACC1
ATN	E43B	E43F	ATN ACC1 ⟶ ACC1	Calcule l'arc tangente de ACC1
GIVAYF	D3ED	D499	A ,Y ⟶ ACC1	Convertit en ACC1 Y = INT BS, A = INT HT ①
QUINT 1	D871	D92C	ACC1 ⟶ # D3, # D4	Convertit ACC1 en # D4 = INT BS, # D3 = INT HT ②
FOUT	EØD1	EØD5	ACC1 ⟶ # 100+	Convertit ACC1 en chaîne, placée à partir de # 100 . ACC1 est détruit

① A priori on ne peut entrer que des nombres compris entre Ø et 32768 soit Ø à # 8ØØØ. Pour A, Y supérieurs on décroit, ainsi A, Y = # 8ØØ1 donne ACC1 = 32767 et A, Y = # FFFF donne ACC1 = 1.

② A priori on obtient un codage "modulo 65536" soit par exemple :
D3, D4 = Ø pour INT (ACC1) = Ø ou 65536 ou — 65536
D3, D4 = # FFFF pour INT (ACC1) = 65535 ou — 1 ou 131071 ou — 65537
D3, D4 = # FFØØ pour INT (ACC1) = — 256 ou 62280



Voici un exemple d'utilisation de ces routines :

Exemple : Diviser 154 par 5, puis mémoriser le résultat à l'adresse 1FFØ pour une utilisation ultérieure, et afficher le résultat.

Adresse	Code	Mnémonique	Commentaires
1FFØ	A9 ØØ	LDA % # Ø	Charge Ø dans l'accumulateur A (% signifie qu'il s'agit d'une constante)
1FØ2	AØ 9A	LDY % # 9A	Charge # 9A soit 154 dans le registre Y
1FØ4	2Ø ED D3	JSR # D3ED	Convertit A, Y en ACC1
1FØ7	2Ø DD DE	JSR # DEDD	Transfère ACC1 dans ACC2
1FØA	A9 ØØ	LDA % # Ø	Charge Ø dans A
1FØC	AØ Ø5	LDY % # Ø5	Charge 5 dans Y
1FØE	2Ø ED D3	JSR # D3ED	Convertit A, Y en ACC1
1F11	2Ø E3 DD	JSR # DDE3	Calcule ACC2/ACC1 —— ACC1
1F14	AØ 1F	LDY % # 1F	Charge # 1F dans Y
1F16	A2 FØ	LDX % # FØ	Charge # FØ dans X
1F18	2Ø A5 DE	JSR # DEA5	Transfère ACC1 dans MEM (X = MEM BS, Y = MEM HT) ici l'adresse de MEM est # 1FFØ
1F1B	2Ø D1 EØ	JSR # EØD1	Transforme ACC1 en chaîne placée à partir de l'adresse # 1ØØ
1F1E	EA	NOP	Instruction muette
1F1F	EA	NOP	
1F2Ø	EA	NOP	
1F21	A2 Ø1	LDX % # Ø1	Charge Ø dans X (qui sert ici de compteur)
1F23	BD ØØ Ø1	LDA # 1ØØ,X	Charge l'octet qui se trouve à l'adresse # 100 + X dans A
1F26	FØ Ø6	BEQ # 1F2E	Branchement à l'adresse # 1F2E si Z = 1 (si résultat nul), ici si A = Ø
1F28	9D 1Ø BE	STA # BE1Ø,X	Charge l'octet qui se trouve dans A à l'adresse # BE1Ø + X (affichage à l'écran à partir de l'adresse # BEØ
1F2B	E8	INX	Incrémente X
1F2C	DØ F5	BNE # 1F23	Branchement à l'adresse # 1F23 si Z = Ø (si résultat non nul), ici si X = Ø
1F2E	6Ø	RTS	Retour (return subroutine)

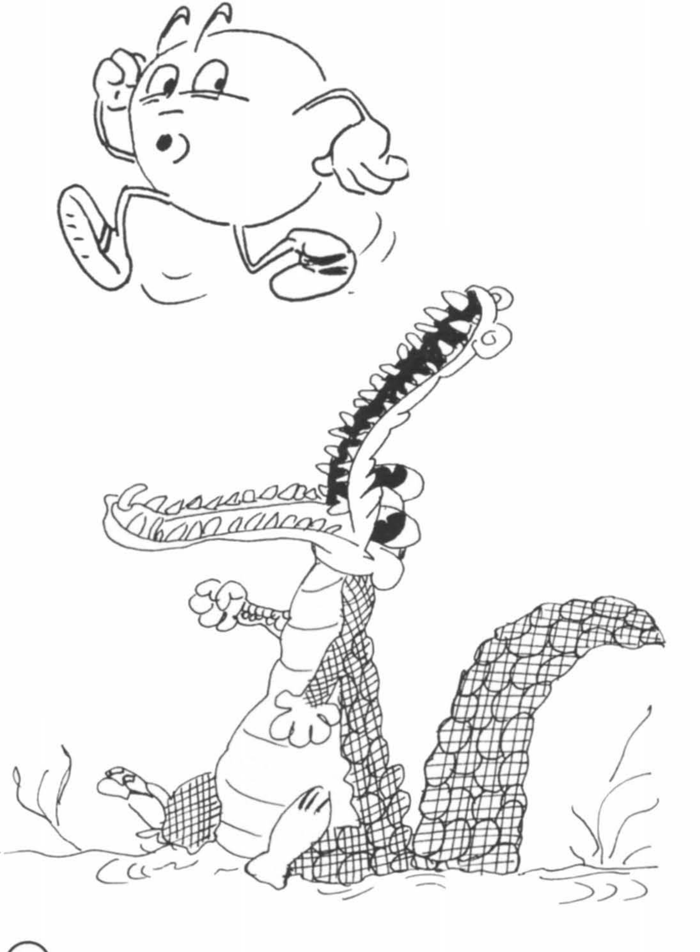
Ce programme est écrit pour ORIC. Pour ATMOS, il convient de changer les adresses des JSR et les remplacer par les adresses correspondantes données dans la liste des routines (cf page 12).

Dans le prochain numéro de MICR'ORIC nous examinerons les variables dimensionnées (ou indicées).

Voici un exemple d'utilisation de ces routines :

Diviser 154 par 5, puis mémoriser le résultat à l'adresse # 1FFØ pour une utilisation ultérieure, et afficher le résultat.

```
1FØØ A9ØØ   LDA %#ØØ
1FØ2 AØ9A   LDY %#9A
1FØ4 2ØEDD3 JSR #D3ED
1FØ7 2ØDDDE JSR #DEDD
1FØA A9ØØ   LDA %#ØØ
1FØC AØØ5   LDY %#Ø5
1FØE 2ØEDD3 JSR #D3ED
1F11 2ØE3DD JSR #DDE3
1F14 AØ1F   LDY %#1F
1F16 A2FØ   LDX %#FØ
1F18 2ØA5DE JSR #DEA5
1F1B 2ØD1EØ JSR #EØD1
1F1E EA     NOP
1F1F EA     NOP
1F2Ø EA     NOP
1F21 A2Ø1   LDX %#Ø1
1F23 BDØØØ1 LDA #Ø1ØØ,X
1F26 FØØ6   BEQ #1F2E
1F28 9D1ØBE STA #BE1Ø,X
1F2B E8     INX
1F2C DØF5   BNE #1F23
1F2E 6Ø     RTS
```





UN TAMPON POUR IMPRIMANTE

(pour ATMOS uniquement)

par Fabrice BROCHE

Les rapports entre l'ORIC et l'imprimante semblent souffrir d'une certaine incompréhension, et ceci pour trois raisons :

① La gestion usuelle du clavier perturbe les entrées / sorties avec l'imprimante.

② Quand l'imprimante imprime, l'ordinateur attend.

③ Quand l'ordinateur calcule, l'imprimante attend.

Nous allons voir comment, en remédiant aux deux derniers points on résoud du même coup le premier.

**** Donnons d'abord quelques brefs rappels sur l'interface parallèle.***

C'est une interface aux normes CENTRONICS dont seulement dix signaux subsistent :

DØ D7 relié directement au BUS de données : R.A.S.

Strobe relié à la broche PB4 du VIA 6522. En le faisant passer à Ø pendant au moins 0,5 µs on signale à l'imprimante d'accepter la donnée qui suit sur le port.

ACK (pour ACKNOWLEDGMENT) relié à la broche CA1 du VIA est utilisée par l'imprimante pour signaler à l'ordinateur qu'elle est à nouveau prête (état bas pendant environ 5 µs).

Voici une procédure standard pour envoyer un caractère :

```
LDA  donnée
STA  #3Ø1    mettre la donnée sur le port
LDA  #3ØØ
AND  %#EF
STA  #3ØØ    mettre la broche PB4 à Ø(Strobe)
ORA  %#2Ø
STA  #3ØØ    puis à 1 pour valider la donnée
```

**** Donnons aussi quelques brefs rappels sur les interruptions.***

Le 65Ø2 possède une ligne de demande d'interruption, l'IRQ. Lorsque cette ligne passe à l'état bas, le 65Ø2 interrompt le programme en cours et se branche à l'adresse contenue en # FFFE-F soit # 244, après avoir sauvegardé le compteur et le registre d'état. Il revient au programme principal en rencontrant l'instruction RTI.

Comment générer des interruptions ?

Le 65Ø2 s'en occupe : il possède un registre d'autorisation d'interruption (en 3ØE). Normalement, il est confirmé pour générer une interruption tous les 1/100e de seconde, il permet ainsi au 65Ø2 de gérer le clavier.

Quelques conseils sur la gestion des interruptions

- Ne pas oublier de sauver les registres A, X, Y si vous les utilisez, puis de les restituer avant la fin de l'interruption.
- Ne pas oublier de remettre à Ø l'indicateur lorsqu'elle a été acquittée.

Pour de plus amples renseignements sur la gestion des interruptions, ainsi qu'à propos de la programmation du 6522 reportez-vous à des ouvrages comme "Applications du 65Ø2" de Rodnay ZAKS aux éditions SYBEX ou "6502 : progammation en assembleur" par L. LEVENTHAL aux Editions Radio.

Revenons à notre problème, la solution consiste en la création d'un tampon imprimante. (Buffer). C'est tout simplement une mémoire qui reçoit les données au rythme **très élevé** de l'ordinateur et qui les envoie au rythme **très lent** de l'imprimante.

Malheureusement un véritable "buffer" coûte cher, mais nous allons simuler son fonctionnement et, ce, par programme. (de manière "soft"dirons ceux qui sont "in").

Nous allons prendre un peu de la mémoire vive de l'ORIC pour servir de mémoire tampon.

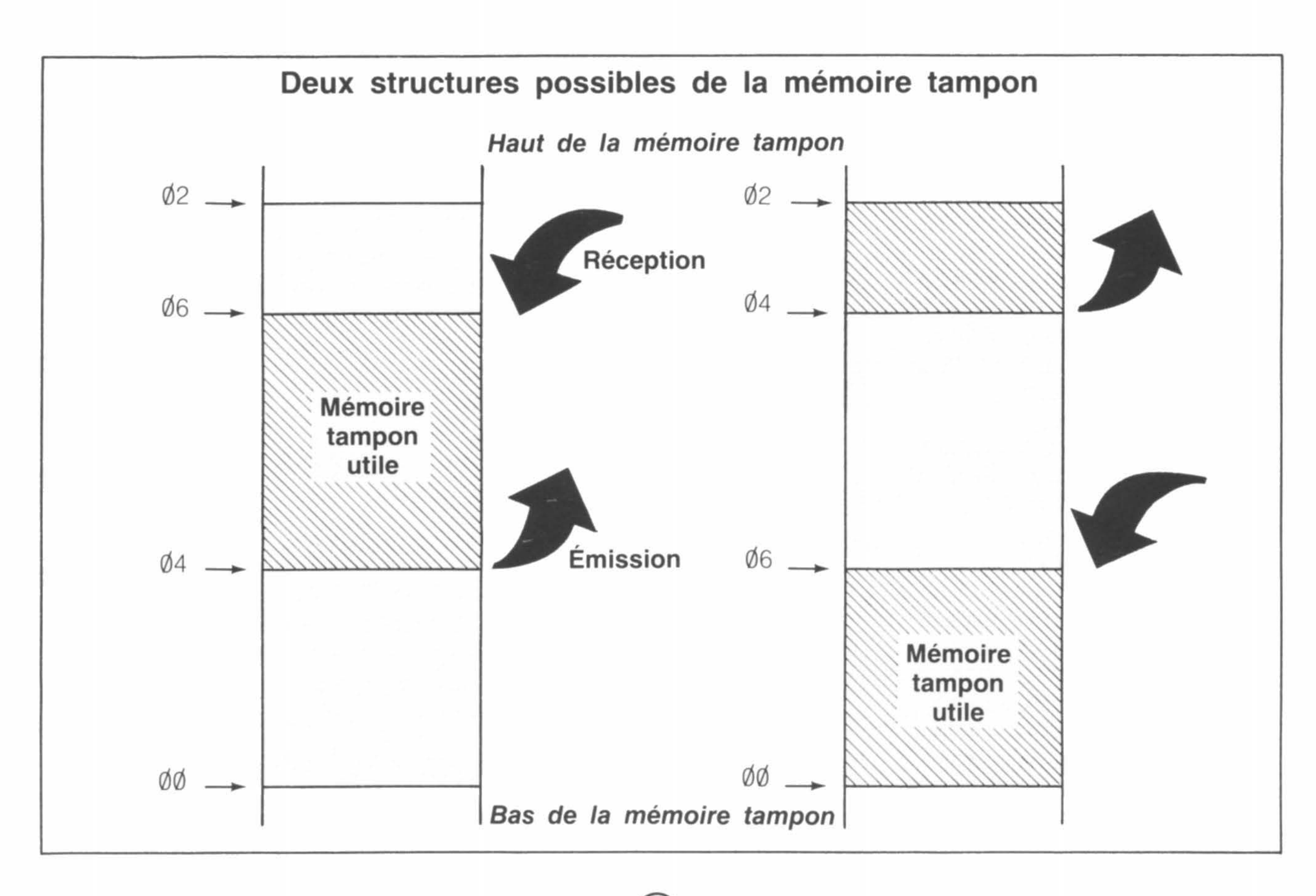
Cette mémoire sera alimentée en entrée par les caractères envoyés en principe à l'imprimante, et, ce, en détournant le **"vecteur imprimante"** situé en # 23F- # 24Ø. Cette **"vectorisation"** n'existe pas sur l'ORIC 1 ce qui explique que ce programme soit réservé à l'ATMOS.

Pour vider cette mémoire nous allons faire en sorte qu'elle envoie une interruption dès qu'elle sera prête.

Le programme d'interruption prendra alors un caractère dans la mémoire tampon et l'enverra à l'imprimante.

Bien entendu il faudra pouvoir distinguer une interruption provoquée par l'imprimante de celle nécessaire à la gestion du clavier : C'est là que nous résolvons notre premier point noir. Le meilleur moyen de pas être gêné par les interruptions est de se servir des interruptions !

Examinons la structure de la mémoire tampon :



Voici le programme désassemblé et commenté

BUFFER IMPRIMANTE ATMOS UNIQUEMENT

```
0400 78     SEI
0401 AD0103 LDA #0301    Mettre à 0 l'indicateur d'interruption sur CA1
0404 A982   LDA %#82
0406 8D0E03 STA #030E    Autoriser interruption sur CA1
0409 A900   LDA %#00
040B A060   LDY %#60
040D 8500   STA #00
040F 8401   STY #01      Début mémoire tampon = #6000
0411 A070   LDY %#70     Fin mémoire tampon = #7000
0413 8502   STA #02
0415 8403   STY #03
0417 A93F   LDA %#3F
0419 A004   LDY %#04
041B 8D4B02 STA #024B    Détourner retour d'interruption vers #43F
041E 20EB04 JSR #04EB
0421 A999   LDA %#99
0423 A004   LDY %#04
0425 8D3F02 STA #023F
0428 8C4002 STY #0240     Détourner vecteur imprimante vers #499
042B 58     CLI
042C A500   LDA #00
042E A401   LDY #01
0430 8504   STA #04
0432 8405   STY #05      Réinitialiser vecteur tampon sortie
0434 8506   STA #06
0436 8407   STY #07
0438 A900   LDA %#00     et entrée
043A 8508   STA #08
043C 8509   STA #09
043E 60     RTS          Longueur tampon = 0
043F 48     PHA
0440 AD0D03 LDA #030D
0443 2902   AND %#02
0445 D009   BNE #0450    Est-ce une interruption sur CA1?
0447 A50A   LDA #0A      Oui...Indic. inter. à 0.Envoi d'un caractère
0449 1003   BPL #044E    Non...L'imprimante est-elle prête?
044B 205604 JSR #0456
044E 68     PLA          Non.Fin
044F 40     RTI          Oui.Envoi à l'imprimante
0450 AD0103 LDA #0301    Retour d'interruption
0453 4C4B04 JMP #044B    Indicateur d'interruption sur CA1 à 0
0456 8A     TXA          Fin
0457 48     PHA
0458 98     TYA          ENVOI DU CARACTERE A L"IMPRIMANTE
0459 48     PHA
045A A508   LDA #08
045C 0509   ORA #09
045E F02A   BEQ #048A    Le tampon est-il vide?
0460 A000   LDY %#00     OUI...FIN
0462 B104   LDA (#04),Y
0464 8D0103 STA #0301    Mettre le caractère sur le BUS de données
0467 AD0003 LDA #0300
046A 29EF   AND %#EF
046C 8D0003 STA #0300    Strobe à 0
```

```
046F 0910   ORA %#10
0471 8D0003 STA #0300    puis à 1   validation
0474 A604   LDX #04
0476 A405   LDY #05
0478 20DA04 JSR #04DA
047B 8604   STX #04
047D 8405   STY #05      Incrémenter pointeur sortie
047F A508   LDA #08
0481 D002   BNE #0485
0483 C609   DEC #09      Décrémenter longueur tampon
0485 C608   DEC #08
0487 A900   LDA %#00     L'imprimante n'est pas prête
0489 2CA980 BIT #80A9    48A...Elle est prête
048C 850A   STA #0A
048E A982   LDA %#82     Autoriser les interruptions au cas où
0490 8D0E03 STA #030E    une routine les interdirait.
0493 68     PLA
0494 A8     TAY
0495 68     PLA
0496 AA     TAX
0497 60_    RTS                    ______________
0498 EA     NOP
0499 867E   STX #7E
049B 847F   STY #7F      ENVOI CARACTERE AU BUFFER
049D 08     PHP          Sauver X et Y
049E 78     SEI
049F A000   LDY %#00
04A1 9106   STA (#06),Y
04A3 A606   LDX #06      Envoi caractère sur la mémoire tampon
04A5 A407   LDY #07
04A7 20DA04 JSR #04DA
04AA 8606   STX #06
04AC 8407   STY #07      Incrémenter pointeur entrée
04AE E608   INC #08
04B0 D002   BNE #04B4
04B2 E609   INC #09      et longueur tampon
04B4 240A   BIT #0A
04B6 3019   BMI #04D1
04B8 E404   CPX #04      Si l'imprimante est prête
04BA D018   BNE #04D4    envoyer tout de suite le caractére
04BC C405   CPY #05
04BE D014   BNE #04D4    Le pointeur entrée rejoint-il le pointeur sortie?
04C0 EA     NOP          Non.Fin
04C1 EA     NOP
04C2 EA     NOP
04C3 EA     NOP
04C4 A902   LDA %#02     Oui
04C6 2C0D03 BIT #030D
04C9 F0F9   BEQ #04C4    Attendre que l'imprimante soit prête
04CB AD0103 LDA #0301    Mettre à 0 l'indicateur d'interruption
04CE EA     NOP
04CF EA     NOP
04D0 EA     NOP
04D1 205604 JSR #0456    Envoyer un caractère à l'imprimante
04D4 A67E   LDX #7E
04D6 A47F   LDY #7F
04D8 28     PLP          Restituer les registres
04D9 60     RTS                    ______________
04DA E8     INX          Incrémenter un pointeur en circuit fermé
04DB D001   BNE #04DE
```

```
04DD C8      INY              Incrémenter Y
04DE E402    CPX #02
04E0 D008    BNE #04EA
04E2 C403    CPY #03
04E4 D004    BNE #04EA        Si en haut de la mémoire tampon
04E6 A600    LDX #00
04E8 A401    LDY #01          alors en bas
04EA 60      RTS                              ____________
04EB 8C4C02  STY #024C        (suite de l'initialisation)
04EE A94C    LDA %#4C         Placer JMP
04F0 8D4A02  STA #024A
04F3 60      RTS
```

Ce programme est très simple dans son principe.

Voici tout de même quelques éclaircissements :

— L'imprimante ne signalant qu'une seule fois fois qu'elle est prête par la broche ACK et si pour une raison ou pour une autre (par exemple, programme chargé alors que l'imprimante est déjà sous tension) l'ORIC attendrait le signal indéfiniment ; il faut pouvoir activer l'imprimante en faisant croire au programme qu'elle envoie le signal indiquant qu'elle est prête. Pour cela il suffit de POKER en 1Ø un nombre supérieur à 128.

— Pour tester facilement si le tampon est plein, le pointeur Ø8 contient la longueur utile. En # 438 on le met à Ø quand il doit être vide.

— Bien entendu, lorsque le tampon est plein, il ne faudra accepter un nouveau caractère qu'après en avoir envoyé un à l'imprimante.

— Dès sa mise en route le programme mémorise l'état de l'imprimante, à l'adresse 1Ø(#ØA).

— Pour lancer le programme un CALL #4ØØ suffit.

— A l'origine, la mémoire choisie pour servir de tampon est située aux adresses # 6ØØØ _# 6FFF Pour modifier ces adresses, ou pour vider le tampon, faire CALL #42C

— L'imprimante est arrêtée pendant les CSAVE et CLOAD. (Il serait possible de faire en sorte qu'elle continue).
il arrive que ces commandes fassent apparaître des caractères inopinés : on ne peut pas l'éviter, cela vient du fait que l'ORIC active le STOBE chaque fois qu'il enclenche le relais de télécommande.



DISK-SEARCH

par Denis SEBBAG

Ce programme, probablement le premier du type sur DOS Oric, permet un examen approfondi d'une disquette, ainsi que la possibilité de récupérer les programmes d'une disquette dont le Directory a été altéré.

La version Basic charge la routine, et la sauve sur disquette (pour ceux qui n'ont pas d'assembleur).

Si vous voulez essayer le programme sans avoir à le sauver, chargez la version machine, et après avoir rebranché le lecteur, tapez : CALL # 4756.

Le programme vous indique pour chaque fichier de la cassette (fichier inscrit dans le directory ou non), piste et secteur de début et de fin. Dans le cas d'un fichier qui a été effacé, le programme indique piste et secteur de début (ce qui permet de retrouver une partie de ce dernier), mais l'indication de secteur et piste de fin n'a plus de valeur, car ils sont replacés par le !DEL au début de la zone libre de la disquette (dans ce cas, on peut aller jusqu'à la dernière piste). En cas de blocage (liens piste-secteur altérés, disquette défectueuse), on peut presser le bouton RESET de l'Oric, puis faire un CALL #4756 pour repartir.

A noter que le chargement de la routine par le programme Basic "Disk-Search" dure assez longtemps (10 s), car la routine occupe tout de même 732 octets.

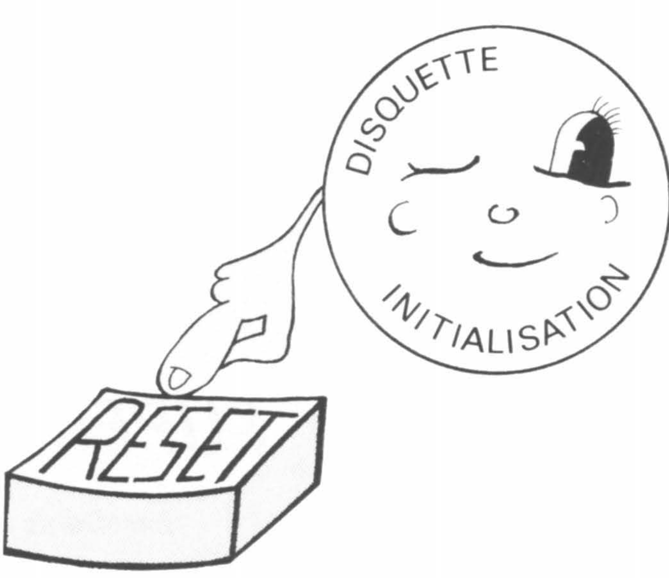
```
 10 REM ¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦
 20 REM ¦¦¦                          ¦¦¦
 30 REM ¦¦¦   *** DISK-SEARCH ***    ¦¦¦
 40 REM ¦¦¦                          ¦¦¦
 50 REM ¦¦¦   Par D.Sebbag     1984  ¦¦¦
 60 REM ¦¦¦                          ¦¦¦
 70 REM ¦¦¦    Oric-1/Oric ATMOS     ¦¦¦
 80 REM ¦¦¦                          ¦¦¦
 90 REM ¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦
 91 REM
 92 REM
 93 REM     ---------------------
 94 REM    : CHARGEMENT  ROUTINE :
 95 REM     ---------------------
 96 REM
 97 REM
 100 CLS:PRINT"CHARGEMENT ROUTINE":POKE#
30E,64:LI=1000
 110 FOR AD=#4756TO#4A32 STEP 5:S=0
 120 FOR J=0 TO 4
 130 READ DT:S=S+DT:POKE AD+J,DT
 140 NEXT
 150 READ CS:IFS<>CS THEN PRINT"Erreur a
 la ligne "LI:POKE#30E,192:END
 160 LI=LI+10:NEXT
 170 POKE#30E,192
 180 REM
 181 REM
 182 REM   --------------------
 183 REM  : SAUVEGARDE ROUTINE :
 184 REM   --------------------
 185 REM
 186 REM
 190 PRINT"Placer une disquette dans le
lecteur  et pressez une touche"
 200 POKE#4FD,1:GETA$
 210 !SAVE"SEARCH.COM",A#4756,E#4A32,AUT
O
 220 IF PEEK(#4FF)=26 THEN PRINT"Disquet
te protegee":GOTO190
 230 REM
 231 REM     ----------------
 232 REM    : DATAS  MACHINE :
 233 REM     ----------------
 234 REM
 1000 DATA #A0,#49,#AD,#07,#C0,605
 1010 DATA #D0,#0B,#A9,#6C,#8D,637
 1020 DATA #20,#49,#A9,#03,#8D,418
 1030 DATA #1C,#49,#C8,#84,#03,436
 1040 DATA #EA,#4C,#F3,#47,#A2,786
 1050 DATA #01,#A9,#2D,#A0,#26,413
 1060 DATA #99,#C1,#BC,#88,#D0,878
 1070 DATA #FA,#A9,#12,#48,#A9,678
 1080 DATA #F6,#A0,#49,#20,#DC,731
 1090 DATA #47,#A0,#4A,#68,#20,441
 1100 DATA #DC,#47,#58,#20,#3B,470
 1110 DATA #D5,#48,#A9,#20,#A0,646
 1120 DATA #26,#99,#82,#BB,#88,644
 1130 DATA #D0,#FA,#68,#C9,#1B,790
 1140 DATA #F0,#46,#60,#85,#D2,749
 1150 DATA #A9,#00,#85,#D1,#A2,673
 1160 DATA #90,#38,#20,#5A,#D4,534
 1170 DATA #22,#DF,#31,#DF,#20,561
 1180 DATA #5A,#D4,#D1,#E0,#D5,948
 1190 DATA #E0,#AD,#01,#01,#AC,571
 1200 DATA #02,#01,#D0,#02,#A0,373
```

```
1210 DATA #20,#60,#A9,#00,#20,329
1220 DATA #5A,#D4,#D0,#F7,#06,763
1230 DATA #F8,#A9,#BB,#85,#13,756
1240 DATA #A9,#A8,#85,#12,#A9,657
1250 DATA #01,#8D,#68,#02,#A9,417
1260 DATA #2F,#4C,#19,#49,#20,253
1270 DATA #5A,#D4,#2F,#F8,#65,698
1280 DATA #F8,#60,#4C,#E6,#48,722
1290 DATA #EA,#EA,#EA,#EA,#EA,1170
1300 DATA #EA,#EA,#EA,#EA,#EA,1170
1310 DATA #EA,#EA,#A0,#49,#A9,870
1320 DATA #10,#8D,#6B,#02,#A9,435
1330 DATA #07,#8D,#6C,#02,#A9,427
1340 DATA #01,#85,#04,#EA,#EA,606
1350 DATA #20,#E6,#04,#A9,#0A,445
1360 DATA #8D,#6A,#02,#20,#5A,371
1370 DATA #D4,#0A,#CC,#CE,#CC,836
1380 DATA #A9,#38,#20,#5A,#D4,559
1390 DATA #ED,#CB,#B0,#CC,#20,852
1400 DATA #6E,#47,#A9,#28,#20,422
1410 DATA #19,#49,#A2,#01,#8E,403
1420 DATA #68,#02,#8E,#02,#C0,442
1430 DATA #CA,#8E,#01,#C0,#EA,771
1440 DATA #20,#24,#D4,#AD,#25,490
1450 DATA #C0,#C9,#FF,#D0,#7A,978
1460 DATA #A5,#04,#20,#9F,#47,431
1470 DATA #8D,#AD,#49,#8C,#AE,701
1480 DATA #49,#AD,#01,#C0,#20,471
1490 DATA #F8,#48,#8D,#BD,#49,723
1500 DATA #8C,#BE,#49,#AD,#02,578
1510 DATA #C0,#20,#F8,#48,#8D,685
1520 DATA #C9,#49,#8C,#CA,#49,689
1530 DATA #AD,#01,#C0,#48,#AD,611
1540 DATA #02,#C0,#48,#AD,#24,475
1550 DATA #C0,#F0,#0F,#8D,#02,590
1560 DATA #C0,#AD,#23,#C0,#8D,733
1570 DATA #01,#C0,#20,#24,#D4,473
1580 DATA #4C,#67,#48,#AD,#01,425
1590 DATA #C0,#20,#F8,#48,#8D,685
1600 DATA #E5,#49,#8C,#E6,#49,745
1610 DATA #AD,#02,#C0,#20,#F8,647
1620 DATA #48,#8D,#F1,#49,#8C,667
1630 DATA #F2,#49,#68,#8D,#02,562
1640 DATA #C0,#68,#8D,#01,#C0,630
1650 DATA #A9,#A5,#A0,#49,#20,599
1660 DATA #5A,#D4,#ED,#CB,#B0,918
1670 DATA #CC,#E6,#04,#A2,#00,600
1680 DATA #A9,#1F,#20,#7C,#47,427
1690 DATA #EA,#EA,#EA,#EA,#EA,1170
1700 DATA #EA,#EA,#EE,#02,#C0,900
1710 DATA #AD,#02,#C0,#C9,#11,585
1720 DATA #D0,#1B,#EE,#01,#C0,666
1730 DATA #AD,#01,#C0,#29,#7F,534
1740 DATA #CD,#13,#C0,#F0,#19,681
1750 DATA #20,#F8,#48,#8D,#A8,661
1760 DATA #BC,#8C,#A9,#BC,#A9,854
1770 DATA #01,#8D,#02,#C0,#20,368
1780 DATA #F8,#48,#8D,#B6,#BC,831
1790 DATA #8C,#B7,#BC,#D0,#40,783
1800 DATA #20,#C1,#47,#A9,#0B,476
1810 DATA #8D,#6A,#02,#58,#4C,413
1820 DATA #22,#DA,#EA,#EA,#EA,954
1830 DATA #EA,#EA,#EA,#48,#29,815
1840 DATA #0F,#09,#30,#C9,#3A,331
1850 DATA #90,#02,#69,#06,#A8,425
1860 DATA #68,#4A,#4A,#4A,#4A,400
1870 DATA #09,#30,#C9,#3A,#90,460
1880 DATA #02,#69,#06,#60,#C9,410
1890 DATA #3A,#90,#02,#69,#06,315
1900 DATA #60,#85,#02,#A0,#07,398
1910 DATA #B1,#02,#99,#77,#02,453
1920 DATA #88,#D0,#F8,#60,#4C,764
1930 DATA #32,#48,#10,#BD,#E8,559
1940 DATA #BC,#D0,#02,#13,#D0,625
1950 DATA #BB,#A8,#BB,#10,#04,562
1960 DATA #1B,#00,#86,#20,#20,225
1970 DATA #20,#20,#2A,#2A,#2A,190
1980 DATA #2A,#2A,#2A,#2A,#20,200
1990 DATA #44,#49,#53,#4B,#2D,344
2000 DATA #53,#45,#41,#52,#43,366
2010 DATA #48,#20,#2A,#2A,#2A,230
2020 DATA #2A,#2A,#2A,#2A,#20,200
2030 DATA #0A,#0A,#0A,#0D,#20,75
2040 DATA #20,#01,#82,#20,#20,227
2050 DATA #2A,#2A,#20,#50,#61,293
2060 DATA #72,#20,#44,#2E,#53,343
2070 DATA #65,#62,#62,#61,#67,497
2080 DATA #20,#20,#81,#60,#20,321
2090 DATA #31,#39,#38,#34,#82,344
2100 DATA #2A,#2A,#20,#0A,#0A,136
2110 DATA #0A,#0D,#20,#20,#20,119
2120 DATA #20,#20,#20,#20,#50,208
2130 DATA #49,#53,#54,#45,#3A,367
2140 DATA #24,#30,#30,#20,#20,196
2150 DATA #20,#53,#45,#43,#54,335
2160 DATA #45,#55,#52,#3A,#24,330
2170 DATA #30,#31,#0A,#0A,#0D,130
2180 DATA #00,#46,#49,#43,#48,282
2190 DATA #49,#45,#52,#20,#31,305
2200 DATA #20,#20,#44,#45,#42,267
2210 DATA #55,#54,#20,#50,#49,354
2220 DATA #53,#54,#45,#3A,#24,330
2230 DATA #30,#30,#20,#53,#45,280
2240 DATA #43,#54,#45,#55,#52,387
2250 DATA #3A,#24,#30,#37,#0D,210
2260 DATA #0D,#20,#20,#20,#20,141
2270 DATA #20,#20,#20,#20,#20,160
2280 DATA #20,#20,#46,#49,#4E,285
2290 DATA #20,#20,#20,#50,#49,249
2300 DATA #53,#54,#45,#3A,#24,330
2310 DATA #32,#37,#20,#53,#45,289
2320 DATA #43,#54,#45,#55,#52,387
2330 DATA #3A,#24,#30,#33,#0D,206
2340 DATA #0A,#00,#20,#20,#20,106
2350 DATA #50,#72,#65,#73,#73,525
2360 DATA #65,#72,#20,#75,#6E,474
2370 DATA #65,#20,#74,#6F,#75,477
2380 DATA #63,#68,#65,#20,#70,448
2390 DATA #6F,#75,#72,#20,#00,374
2400 DATA #63,#6F,#6D,#6D,#65,529
2410 DATA #6E,#63,#65,#72,#20,456
2420 DATA #20,#20,#00,#63,#6F,274
2430 DATA #6E,#74,#69,#6E,#75,558
2440 DATA #65,#72,#00,#C0,#BC,595
2450 DATA #13,#00,#00,#00,#00,19
2460 DATA #80,#BB,#1B,#00,#00,342
```

INITIALISATION

par Denis SEBBAG

Ce programme permet d'introduire sur une disquette une présentation, à chaque appui sur le bouton RESET du lecteur, ou une mise en marche de l'ensemble.



Cette présentation provoque le passage en encre blanche sur fond noir (plus agréable pour les yeux, supprime le déclic clavier, et sur Oric Atmos augmente la vitesse de répétition. Puis la date d'initialisation de la disquette et le nom de l'auteur sont affichés. Ensuite, sur simple pression de la touche D, le Directory (répertoire) est affiché (ou n'est pas affiché en pressant une autre touche. Pour lancer le programme, tapez simplement RUN, puis le programme vous demande les indications nécessaires, puis sauve la présentation sur disquette (sous le nom "BOOTUP.COM" en l'occurence).

Remarque : Pour que la disquette soit effectivement initialisée (départ à la mise sous tension), il faut qu'elle contienne au préalable le SYSTEM. DOS (un !COPY suffit, si ce n'est pas le cas).

```
 10 REM::::::::::::::::::::::::::::::::
 11 REM:::                          :::
 12 REM:::                          :::
 13 REM::: *** INITIALISATION ***   :::
 14 REM:::        De disquettes     :::
 15 REM:::                          :::
 16 REM:::                          :::
 17 REM:::  Par D.Sebbag  13/05/84  :::
 18 REM:::                          :::
 19 REM:::     Oric-1/Oric ATMOS    :::
 20 REM:::                          :::
 21 REM::::::::::::::::::::::::::::::::
 22 REM
 23 REM    ------------------------
 24 REM   :                        :
 25 REM   :  CHARGEMENT MACHINE    :
 26 REM   :                        :
 27 REM   :                        :
 28 REM    ------------------------
 29 REM
 30 REM
 35 CLS:PRINT"CHARGEMENT ROUTINE":POKE#3
0E,64:LI=1000
 40 FORI=#400TO#46FSTEP5:S=0:FORJ=0TO4:R
EAD DT:IFI+J>#46F THEN NEXT:GOTO50
 45 S=S+DT:POKEI+J,DT:NEXT
 50 READ CS:IFS<>CS THENPRINT"Erreur a l
a ligne "LI:GOTO230
 55 LI=LI+10
 60 NEXT:POKE#30E,192
 70 REM
 71 REM    ---------------------
 72 REM   :                     :
 73 REM   :    DATE ET NOM      :
 74 REM   :                     :
 75 REM    ---------------------
 76 REM
 78 POKE#31,40
 80 PRINT:PRINT:PRINT"Date d'initialisat
ion":INPUT"(JJ/MM/AA) ";DA$
 90 IFLEN(DA$)<>8THEN80
 100 FORI=1TO8:POKE#447+I,ASC(MID$(DA$,I
,1)):NEXT
 110 PRINT"Nom de l'auteur"
 120 INPUT"(12 lettres max) ";NO$
 130 IFLEN(NO$)>12THEN110
 140 FORI=1TOLEN(NO$):POKE#455+I,ASC(MID
$(NO$,I,1)):NEXT
 150 REM
 151 REM    ----------------------
 152 REM   :                      :
 153 REM   :   INIT. DISQUETTE    :
 154 REM   :                      :
 155 REM    ----------------------
 156 REM
 160 PRINT:PRINT"Placer la disquette a i
nitialiser dansle lecteur , et ne la ";
 170 PRINT"retirer que     lorsque le vo
yant est eteint .";
 180 PRINT:PRINT:PRINT"Presser une touch
e pour commencer ";:GETA$
 190 POKE#4FD,1:!DEL"BOOTUP.COM"
```

```
 195 IF PEEK(#4FF)=26 THENPRINT:PRINT"Er
reur:disquette protegee":GOTO180
 200 !SAVE"BOOTUP.COM",A#400,E#46F,AUTO
 205 IF PEEK(#4FF)=26 THEN 195
 210 PRINT:PRINT:PRINT"R pour recommence
r ";:GETA$:IFA$<>"R"ANDA$<>"r"THENEND
 220 PRINT:PRINT:PRINT"Memes donnees (O/
N) ?";:GETA$:IFA$="O"ORA$="o"THEN180ELSE
80
 230 POKE#30E,192
 300 REM
 310 REM    ---------------------
 320 REM    :                   :
 330 REM    :  DATAS MACHINE    :
 340 REM    :                   :
 350 REM    ---------------------
 360 REM
 1000 DATA #A9,#0B,#8D,#6A,#02,429
 1010 DATA #A9,#02,#8D,#4F,#02,393
 1020 DATA #A9,#10,#8D,#4E,#02,406
 1030 DATA #A9,#10,#8D,#6C,#02,436
 1040 DATA #A9,#07,#8D,#6B,#02,426
 1050 DATA #20,#5A,#D4,#0A,#CC,548
 1060 DATA #CE,#CC,#A9,#39,#A0,796
 1070 DATA #04,#20,#5A,#D4,#ED,575
 1080 DATA #CB,#B0,#CC,#58,#20,703
 1090 DATA #3B,#D5,#C9,#44,#D0,749
 1100 DATA #03,#4C,#A5,#E1,#20,501
 1110 DATA #22,#DA,#49,#6E,#69,540
 1120 DATA #74,#69,#61,#6C,#69,531
 1130 DATA #73,#65,#65,#20,#6C,457
 1140 DATA #65,#20,#20,#20,#2F,244
 1150 DATA #20,#20,#2F,#20,#20,175
 1160 DATA #0A,#0D,#70,#61,#72,346
 1170 DATA #20,#20,#20,#20,#20,160
 1180 DATA #20,#20,#20,#20,#20,160
 1190 DATA #20,#20,#20,#0D,#0A,119
 1200 DATA #44,#20,#70,#6F,#75,440
 1210 DATA #72,#20,#44,#49,#52,369
 1220 DATA #20,#00,#00,#00,#00,32
```

Pour ATMOS afin d'éviter des ennuis avec l'imprimante :
DOKE # 256, # 285Ø : IF PEEK (# FFF9) = 1 THEN CALL # C83E

SUPER D O S

par Denis SEBBAG

Ce programme donne à votre D O S de nouvelles commandes, et augmente la capacité de la disquette (20 K-octets en plus).

Une fois chargé, taper RUN. Le programme vous demande d'introduire dans le lecteur une disquette contenant "l'ancien" D O S (important). Le programme va copier, puis modifier ce DOS. Ensuite, il vous demande de placer dans le lecteur la disquette où vous souhaitez copier le nouveau DOS (attention, si la disquette contenait l'ancien DOS, il est automatiquement effacé). Après pression d'une touche, vous aurez en votre possession une disquette capable de contenir 180 Koctets, tout en conservant toutes les commandes !Backup et !Copy.

Différences entre le DOS.V1.1 et sa version modifiée : le nouveau DOS reconnaît les noms de fichiers contenant des mots clés du Basic, même lorsqu'ils sont tapés sans la commande LOAD. Exemple : si vous avez un fichier dont le nom est "ZORGON.COM" par exemple (nom qui contient le mot clé du basic "OR"), il suffira de taper ! ZORGON pour le charger, et le message "Invalid filename : ERROR" n'apparaîtra pas.

Une nouvelle commande : d'origine, les disquettes du DOS Oric sont formattées en 40 pistes, 16 secteurs. Une nouvelle commande, !CONF, permet à ce DOS de passer en 44 pistes, ce qui fournit un gain de 20 Koctets par disquettes. Pour passer en 44 pistes, il suffit de taper !CONF 1.



A partir de ce moment là, tout formattage d'une disquette lui donnera 44 pistes (7Ø2 secteurs libres au lieu de 638), et la commande !BACKUP copiera les 44 pistes. Pour cette raison, lors d'un BACKUP, la taille de la mémoire tampon est augmentée et arrive jusqu'à l'adresse # 5ØØ, de sorte qu'après un BACKUP, il faut faire un POKE # 5ØØ,Ø pour utiliser un programme basic. Pour revenir à la configuration normale (40 pistes), taper !CONFØ. Cela est nécessaire par exemple pour faire un BACKUP d'une disquette formattée en 40 pistes. A noter, pour faire la distinction entre les deux DOS : au démarrage, le message "Oric DOS V1.1S" est affiché (S pour special).

```
 10 REM:::::::::::::::::::::::::::::::::
 11 REM:::                           :::
 12 REM:::    *** SUPER-DOS ***      :::
 13 REM:::                           :::
 14 REM:::                           :::
 15 REM:::    Par D.Sebbag  ` 1983   :::
 16 REM:::                           :::
 17 REM:::                           :::
 18 REM:::    Oric-1   / Oric ATMOS  :::
 19 REM:::                           :::
 20 REM:::::::::::::::::::::::::::::::::
 21 REM
 22 REM
 23 REM   -----------------------------
 24 REM   :                           :
 25 REM   :    CHARGEMENT   DOS V1.1  :
 26 REM   :                           :
 27 REM   -----------------------------
 28 REM
 30 HIMEM#7300
 35 CLS
 40 PRINT"Placer dans le lecteur une dis
quette  contenant le DOS"
 50 PRINT:PRINT"Presser une touche pour
commencer ";:GETA$:PRINT
 60 POKE#4FD,1
 70 !LOAD"SYSTEM.DOS",N:IFPEEK(#4FF)=1TH
EN PRINT:PRINT"Pas de DOS":GOTO40
 80 REM
 90 REM   ----------------------------
 91 REM   :                          :
 92 REM   :    MODIFICATION DU DOS   :
 93 REM   :                          :
 94 REM   ----------------------------
 95 REM
 100 POKE#30E,64
 104 REM
 105 REM       ------------
 106 REM       : ROUTINE 1 :
 107 REM       ------------
 108 REM
 110 FORAD=#7300TO#7355:READ DT:S=S+DT:P
OKEAD,DT:NEXT
 120 IFS<>10811THENPOKE#30E,192:PRINT"ER
REUR DANS LA ROUTINE 1":END
 124 REM
 125 REM       ------------
 126 REM       : ROUTINE 2 :
 127 REM       ------------
 128 REM
 130 FOR AD=#A004 TO #A02E:READ DT:S1=S1
+DT:POKEAD,DT:NEXT
 140 IF S1<>5649 THEN POKE#30E,192:PRINT
"ERREUR DANS LA ROUTINE 2":END
 144 REM
 145 REM       ------------
 146 REM       : ROUTINE 3 :
 147 REM       ------------
 148 REM
 150 FOR AD=#400 TO #44B:READ DT:S2=S2+D
T:POKE AD,DT:NEXT
 160 IFS2<>7471THEN PRINT"ERREUR DANS LA
 ROUTINE 3":POKE#30E,192:END
 161 CALL#400
 162 REM
 163 REM    ------------------------
 164 REM    : ROUTINE 4 ET DIVERS :
 165 REM    ------------------------
 166 REM
 167 DOKE#80AC,#20:DOKE#80AE,#EAD3:POKE#
73EB,#52:POKE#73C2,#47
 168 DOKE#791E,#D358:FORAD=#73F0TO#73FE:
READ DT:POKE AD,DT:NEXT
 169 POKE#73D7,#D8:DOKE#790B,#D3C3:DOKE#
790F,#D3AE:DOKE#7918,#D3D7
 170 DOKE#73AA,#B443:POKE#73AC,#46:POKE#
78FF,#14:POKE#9FDD,#53
 175 FORAD=#7848TO#7893:READ DT:POKE AD,
DT:S3=S3+DT:NEXT
 177 POKE#30E,192:IFS3<>9503THEN PRINT"E
RREUR DANS LA ROUTINE 4":END
 181 REM
 182 REM   ------------------------------
 183 REM   :                            :
 184 REM   :  SAUVEGARDE NOUVEAU DOS    :
 185 REM   :                            :
 186 REM   ------------------------------
 187 REM
 190 PRINT:PRINT"Placez dans le lecteur
la disquette oucopier le DOS "
 200 PRINT:PRINT"Presser une touche pour
 commencer";:GETA$:PRINT:!DEL"SYSTEM.DOS
"
 210 !SAVE"SYSTEM.DOS",A#7300,E#A030,T#A
000
 220 IFPEEK(#4FF)=9THENPRINT:PRINT"Disqu
ette protegee":GOTO190
 230 PRINT:PRINT"Une autre sauvegarde (O
/N) ?";:GETA$:IFA$="O"ORA$="o"THEN 190
 240 END
 900 REM
 910 REM   ---------------------------
 911 REM   :                         :
 912 REM   :      DATAS MACHINE      :
 913 REM   :                         :
 914 REM   ---------------------------
 915 REM
 916 REM   ROUTINE 1
 917 REM
 1000 DATA #C9,#7B,#90,#4A,#C9,#80
 1010 DATA #90,#48,#38,#E9,#7F,#86
 1020 DATA #B9,#84,#B8,#AA,#A0,#00
 1030 DATA #A9,#E9,#85,#18,#A9,#C0
 1040 DATA #85,#19,#CA,#F0,#0E,#E6
 1050 DATA #18,#D0,#02,#E6,#19,#20
 1060 DATA #70,#04,#10,#F5,#4C,#1A
 1070 DATA #D3,#A6,#B9,#C8,#20,#70
 1080 DATA #04,#08,#29,#7F,#20,#98
 1090 DATA #E0,#F0,#18,#28,#30,#09
 1100 DATA #9D,#2C,#C1,#E8,#CE,#41
 1110 DATA #C1,#D0,#E8,#A4,#B8,#85
 1120 DATA #B8,#68,#68,#A5,#B8,#60
```

```
1130 DATA #68,#68,#60,#28,#A4,#B8
1140 DATA #58,#60
1145 REM
1146 REM   ROUTINE 2
1147 REM
1150 DATA #A0,#2D,#78,#BD,#00,#73
1160 DATA #9D,#00,#D3,#4D,#2F,#A0
1170 DATA #8D,#2F,#A0,#E8,#D0,#F1
1180 DATA #EE,#09,#A0,#EE,#0C,#A0
1190 DATA #88,#D0,#E8,#A2,#0F,#BD
1200 DATA #F0,#73,#9D,#70,#04,#CA
1210 DATA #10,#F7,#A9,#28,#85,#31
1220 DATA #60
1224 REM
1225 REM   ROUTINE 3
1226 REM
1230 DATA #A9,#D8,#85,#01,#A9,#73
1240 DATA #85,#03,#A9,#48,#85,#00
1250 DATA #A9,#58,#85,#02,#A0,#51
1260 DATA #20,#39,#04,#A9,#99,#85
1270 DATA #00,#A9,#AD,#85,#02,#A0
1280 DATA #13,#20,#39,#04,#A9,#AE
1290 DATA #85,#00,#A9,#C3,#85,#02
1300 DATA #A0,#19,#20,#39,#04,#A9
1310 DATA #C2,#85,#00,#A9,#D8,#85
1320 DATA #02,#A0,#12,#A9,#00,#20
1330 DATA #E6,#04,#B1,#00,#91,#02
1340 DATA #88,#10,#F9,#A9,#02,#20
1350 DATA #E6,#04,#58,#60
1360 REM
1370 REM   ROUTINE 4
1375 REM
1380 DATA #A9,#02,#20,#E6,#04,#B1
1390 DATA #18,#48,#A9,#00,#20,#E6
1400 DATA #04,#68,#60
1410 DATA #20,#5D,#D4,#90,#3E,#20
1420 DATA #6F,#D4,#AD,#45,#C1,#C9
1430 DATA #02,#B0,#2F,#AA,#18,#69
1440 DATA #0A,#8D,#D2,#F8,#8D,#2A
1450 DATA #F9,#BD,#92,#D8,#8D,#CD
1460 DATA #F8,#8D,#25,#F9,#BD,#90
1470 DATA #D8,#AA,#A0,#03,#B9,#13
1480 DATA #C0,#F0,#04,#8A,#99,#13
1490 DATA #C0,#88,#10,#F4,#20,#5A
1500 DATA #D4,#E2,#00,#E2,#00,#D0
1510 DATA #F7,#60,#A2,#0B,#4C,#1B
1520 DATA #D4,#A2,#01,#4C,#1B,#D4
1530 DATA #28,#2C,#14,#05
```

INFORMATION

Monsieur Denis TAIEB qui a assuré la présidence de la société A.S.N. Diffusion de 1979 à 1983 a suivi le développement d'ORIC en France et, avec son équipe, a permis le succès que l'on connaît.

Depuis le mois de juin 1984, il fait partie du Conseil d'Administration de la société ORIC PRODUCT INTERNATIONAL et de son groupe financier.

L'action qu'il portera dans les prochains mois sera une pénétration commerciale à travers l'Europe par la création d'une structure dont le siège sera à Paris.

Sa mission sera aussi d'analyser les conditions d'implantation d'une unité d'assemblage en France et de conseiller la firme ORIC PRODUCT INTERNATIONAL au cours de cette opération.

Monsieur Denis TAIEB espère, par son action, renforcer la pénétration d'ORIC à travers l'Europe pour atteindre 11 % du parc total des microordinateurs familiaux. Les estimations de ce parc sont de 3,5 à 4 millions de machines à fin 1984.

UN MERGE POUR ORIC-1



par Fabrice BROCHE

Il manquait un "merge" sur votre ORIC-1, à tel point que l'ATMOS vous en propose un.

Afin que les possesseurs d'ORIC ne se sentent pas frustrés, voici un "MERGE" avec un "plus".

Il mélangera les deux programmes si les numéros de lignes sont distincts. En cas de numéro identique la nouvelle ligne remplace l'ancienne.

UTILISATION :

!CLOAD NORMAL, "Nom de fichier" [,S]
pour fusionner le programme BASIC de la cassette avec celui présent en mémoire vive.

!CLOAD adresse, "Nom de fichier" [,S]
pour forcer le programme (BASIC ou non) à se charger à l'adresse demandée.

EN OUTRE :

?&(Ø) retourne l'adresse décimale du début de programme, enregistrée sur la bande.

?&(1) retourne l'adresse décimale de la fin du programme, enregistrée sur la bande.

?&(2) retourne l'adresse décimale du début de programme demandé.

?&(3) retourne l'adresse décimale de la fin du programme telle qu'elle résulte du chargement effectué.

Voilà de quoi rendre jaloux les possesseurs d'Atmos !

```
B200 A959   LDA x#59        B200        INITIALISATION
B202 A0B2   LDY x#B2
B204 8DF502 STA #02F5
B207 8CF602 STY #02F6       B207        Détourner "!"
B20A A940   LDA x#40
B20C A0B2   LDY x#B2
B20E 8DFC02 STA #02FC
B211 8CFD02 STY #02FD       B211        Détourner "&"
B214 A91B   LDA x#1B
B216 A0B2   LDY x#B2
B218 4CEDCB JMP #CBED       B218        Afficher message

B21B 0C 0A 09 09 84 43 48 41 .....CHA       B21B-B23D Message
B223 49 4E 41 47 45 53 82 8C INAGES..
B22B 60 20 52 69 70 65 6C 6C ` Ripell
B233 65 20 53 6F 66 74 77 61 e Softwa
B23B 72 65 0A 0A 00          re...
```

B240 2067D8 JSR #D867	**B240**	& # 33-4 = Valeur de l'accumulateur décimale a virgule flottante)
B243 A533 LDA #33		
B245 D009 BNE #B250	**B245**	Sinon nul
B247 ACFCB1 LDY #B1FC		
B24A ADFDB1 LDA #B1FD		
B24D 4CD5D8 JMP #D8D5	**B24D**	A1 ——▸ ACC1
B250 ACFEB1 LDY #B1FE		
B253 ADFFB1 LDA #B1FF		
B256 4CD5D8 JMP #D8D5	**B256**	A1 ——▸ ACC1
B259 A9B6 LDA %#B6		
B25B 20DBCF JSR #CFDB	**B25B**	! Demander CLOAD
B25E 20E800 JSR #00E8		
B261 C983 CMP %#83		
B263 D00B BNE #B270	**B263**	Suivi de NORMAL ?
B265 20E200 JSR #00E2	**B265**	Si oui, le sauter
B268 A59C LDA #9C		
B26A A49D LDY #9D	**B26A**	Charger le programme a la fin du programme en cours et indiquer MERGE
B26C A280 LDX %#80		
B26E D009 BNE #B279		
B270 209DE7 JSR #E79D	**B270**	Saisir adresse
B273 A533 LDA #33		
B275 A434 LDY #34		
B277 A200 LDX %#00		
B279 857E STA #7E		
B27B 847F STY #7F		
B27D 860A STX #0A		
B27F 20D9CF JSR #CFD9	**B27F**	Demander "," (virgule)
B282 2025E7 JSR #E725	**B282**	Syntaxe CLOAD
B285 20CAE6 JSR #E6CA	**B285**	Initialisation du VIA
B288 2063E5 JSR #E563	**B288**	Effacer ligne Ø
B28B A903 LDA %#03		
B28D A0E5 LDY %#E5		
B28F 2076E5 JSR #E576	**B28F**	Afficher SEARCHING
B292 2096E6 JSR #E696	**B292**	Reconnaître bande amorce
B295 2030E6 JSR #E630		
B298 C924 CMP %#24		
B29A D0F9 BNE #B295	**B29A**	Attendre début programme
B29C A209 LDX %#09		
B29E 2030E6 JSR #E630		
B2A1 955D STA #5D,X		
B2A3 CA DEX		
B2A4 D0F8 BNE #B29E	**B2A4**	Changer en-tête
B2A6 240A BIT #0A		
B2A8 1004 BPL #B2AE		
B2AA A564 LDA #64		
B2AC D0DA BNE †B288	**B2AC**	Si MERGE et Machine ——▸ SEARCHING
B2AE A55F LDA #5F		
B2B0 A460 LDY #60		
B2B2 8DFCB1 STA #B1FC		
B2B5 8CFDB1 STY #B1FD	**B2B5**	Sauver début programme
B2B8 38 SEC		

```
B2B9 A561    LDA #61
B2BB 8DFEB1  STA #B1FE
B2BE E55F    SBC #5F          B2BE       Sauver fin de programme
B2C0 48      PHA
B2C1 A562    LDA #62
B2C3 8DFFB1  STA #B1FF
B2C6 E560    SBC #60
B2C8 A8      TAY
B2C9 68      PLA
B2CA 18      CLC
B2CB 657E    ADC #7E
B2CD 8561    STA #61
B2CF 98      TYA
B2D0 657F    ADC #7F
B2D2 8562    STA #62          B2D2       Et positionner nouvelle fin programme
B2D4 A57E    LDA #7E
B2D6 A47F    LDY #7F

B2D8 855F    STA #5F
B2DA 8460    STY #60          B2DA       Et nouveau debut
B2DC 2030E6  JSR #E630
B2DF 9549    STA #49,X
B2E1 F003    BEQ #B2E6
B2E3 E8      INX
B2E4 D0F6    BNE #B2DC        B2E4       Charger nom programme
B2E6 20F0E6  JSR #E6F0        B2E6       Vérifier
B2E9 8A      TXA
B2EA D0AC    BNE #B298        B2EA       Si différent ——► SEARCHING
B2EC 2063E5  JSR #E563
B2EF A912    LDA x#12         B2EF       Effacer ligne Ø
B2F1 A0E5    LDY x#E5
B2F3 2076E5  JSR #E576        B2F3       Afficher LOADING
B2F6 206EE5  JSR #E56E        B2F6       Et nom de programme
B2F9 A55F    LDA #5F
B2FB A460    LDY #60

B2FD 8533    STA #33
B2FF 8434    STY #34          B2FF       Initialiser pointeur chargement
B301 A000    LDY x#00
B303 2030E6  JSR #E630        B303       Charger 1 octet
B306 B009    BCS #B311
B308 9133    STA (#33),Y      B308       Et l'envoyer en mémoire
B30A 2054E5  JSR #E554        B30A       Vérifier si fin
B30D 90F4    BCC #B303
B30F B010    BCS #B321
B311 4C4AE5  JMP #E54A        B311       LOAD ABORTED
                                         (chargement non réussi)

B314 10 07 57 6F 72 6B 69 6E ..Workin    B314-B320  Message "WORKING"
B31C 67 2E 2E 2E 00          g

B321 2004E8  JSR #E804        B321       Initialisation du VIA
B324 240A    BIT #0A
B326 3010    BMI #B338        B326       Si pas MERGE
B328 A564    LDA #64
B32A D00B    BNE #B337        B32A       Et non BASIC, alors fin
```

```
B32C A561   LDA #61
B32E A462   LDY #62
B330 859A   STA #9A           B330  Initialiser pointeur fin de programme et
                                    adresse de liaison
B332 849C   STY #9C
B334 206FC5 JSR #C56F
B337 60     RTS
B338 78     SEI               B338  MERGE
B339 A914   LDA %#14
B33B A0B3   LDY %#B3
B33D 2076E5 JSR #E576         B33D  Afficher "WORKING"

B340 A55F   LDA #5F
B342 A460   LDY #60
B344 850A   STA #0A
B346 840B   STY #0B           B346  ØA va décrire programme à insérer
B348 A001   LDY %#01
B34A B10A   LDA (#0A),Y
B34C D007   BNE #B355         B34C  Si fin ,
B34E 2063E5 JSR #E563         B34E  Effacer ligne Ø
B351 58     CLI
B352 4CB5C4 JMP #C4B5         B352  Et saut à l'interpréteur
B355 C8     INY
B356 B10A   LDA (#0A),Y
B358 8533   STA #33
B35A C8     INY
B35B B10A   LDA (#0A),Y
B35D 8534   STA #34           B35D  N° de ligne ——► # 33-4
B35F A200   LDX %#00

B361 C8     INY
B362 B10A   LDA (#0A),Y
B364 9535   STA #35,X
B366 F003   BEQ #B36B
B368 E8     INX
B369 D0F6   BNE #B361         B369  Recopier ligne dans tampon du clavier
B36B 38     SEC
B36C 98     TYA
B36D 650A   ADC #0A
B36F 850A   STA #0A
B371 9002   BCC #B375
B373 E60B   INC #0B           B373  ØA pointe sur ligne suivante
B375 18     CLC
B376 8A     TXA
B377 6905   ADC %#05
B379 8526   STA #26           B379  Longueur ligne ——► # 26
B37B 20DEC6 JSR #C6DE         B37B  Trouver adresse ligne

B37E 9044   BCC #B3C4
B380 A001   LDY %#01
B382 B1CE   LDA (#CE),Y       B382  RECOPIE ROUTINE D'INSERTION DU
                                    BASIC (jusqu'au bout)
B384 8592   STA #92
B386 A59C   LDA #9C
B388 8591   STA #91
B38A A5CF   LDA #CF
```

```
B38C 8594   STA #94
B38E A5CE   LDA #CE
B390 88     DEY
B391 F1CE   SBC (#CE),Y
B393 18     CLC
B394 659C   ADC #9C
B396 859C   STA #9C
B398 8593   STA #93
B39A A59D   LDA #9D
B39C 69FF   ADC ×#FF

B39E 859D   STA #9D
B3A0 E5CF   SBC #CF
B3A2 AA     TAX
B3A3 38     SEC
B3A4 A5CE   LDA #CE
B3A6 E59C   SBC #9C
B3A8 A8     TAY
B3A9 B003   BCS #B3AE
B3AB E8     INX
B3AC C694   DEC #94
B3AE 18     CLC
B3AF 6591   ADC #91
B3B1 9003   BCC #B3B6
B3B3 C692   DEC #92
B3B5 18     CLC
B3B6 B191   LDA (#91),Y
B3B8 9193   STA (#93),Y

B3BA C8     INY
B3BB D0F9   BNE #B3B6
B3BD E692   INC #92
B3BF E694   INC #94
B3C1 CA     DEX
B3C2 D0F2   BNE #B3B6
B3C4 2033C7 JSR #C733
B3C7 206FC5 JSR #C56F
B3CA EA     NOP
B3CB EA     NOP
B3CC EA     NOP
B3CD EA     NOP
B3CE 18     CLC
B3CF A59C   LDA #9C
B3D1 85C9   STA #C9
B3D3 6526   ADC #26
B3D5 85C7   STA #C7

B3D7 A49D   LDY #9D
B3D9 84CA   STY #CA
B3DB 9001   BCC #B3DE
B3DD C8     INY

B3DE 84C8   STY #C8
B3E0 20F8C3 JSR #C3F8
B3E3 A5A0   LDA #A0
B3E5 A4A1   LDY #A1
B3E7 859C   STA #9C
B3E9 849D   STY #9D
B3EB A426   LDY #26
B3ED 88     DEY
B3EE B93100 LDA #0031,Y
B3F1 91CE   STA (#CE),Y
B3F3 88     DEY
B3F4 10F8   BPL #B3EE
B3F6 2033C7 JSR #C733
B3F9 206FC5 JSR #C56F
B3FC 4C48B3 JMP #B348
B3FF EA     NOP
```

MODE D'EMPLOI :

Si vous ne connaissez pas encore le langage machine, il vous suffit d'écrire un programme de chargement des données ainsi conçu :

```
10  FOR I = # B200 TO # B3FF
20  READ DT
30  POKE I , DT
40  NEXT
100 DATA # A9, # 59, # A0, # B2, # 8D, # F5
    etc... jusqu'à # B3, # EA
```

Par prudence sauver le programme. L'exécuter une fois. On peut alors sauver le bloc mémoire de # B200 à # B3FF pour une utilisation ultérieure.

L'emploi se fait par CALL # B200.



B.D.
CAPTAIN TANEX
YUK !
LES MURS SONT COUVERTS DE BAVE.
SSSSS
CETTE IMMONDE FANGE BRÛLE !
PENDANT CE TEMPS... LA-BAS.
ORIC-Ω DECHARGE UN ELECTRONVOLT
MAIS MEME UN MILLIARD DE MÉGAVOLTS N'AGIRAIENT PAS...
LE TIROK A PROJETTE SA TROMPE MORTELLE.
CAPTAIN TANEX & ORIC-Ω
THUP
PEU APRÈS
MAINTENANT SUS A L'INTRUS.
SERAIT-CE L'ANTRE DU TIROKA ?
SÛR QUE CE N'EST PAS LE TRÉSOR !
MAIS SI CE N'EST PAS...
JE N'AI PLUS LE TEMPS DE ....
PEUT PLUS M'ÉCHAPPER MAINTENANT

Trucs et Astuces

par Fabrice BROCHE

MICR'ORIC se fait un plaisir de vous indiquer des méthodes qui vous permettront de programmer de plus en plus aisément sur votre système ORIC. L'humour n'est pas exclus mais les amateurs ne s'y tromperont pas, ils trouveront ici des renseignements très précieux.

PRINT = LPRINT

L'ORIC contient une variable système qui indique à tout moment quel doit être le terminal de visualisation, à savoir **écran** ou **imprimante :** c'est l'octet # 2F1, dont le bit 7 indique, lorsqu'il est à Ø la sortie vers l'écran et lorsqu'il est monté à 1 la sortie sur imprimante. Et ceci pour tout ordre PRINT, INPUT "" etc., en excluant PLOT et évidemment les POKES dans la mémoire écran.

Donc, normalement le contenu de l'octet # 2F1 est inférieur à 128 puisqu'on écrit sur l'écran. Mais écrivons POKE # 2F1, 128 suivi d'un <Return>. A l'exécution, tout ce qui, dans le programme s'affichait à l'écran, ira s'écrire à l'imprimante : PRINT deviendra LPRINT, LIST deviendra LLIST en quelque sorte.

Ceci est dû tout simplement au fait que l'ordre LPRINT par exemple ne fait que forcer à 1 le 7e bit de # 2F1, et le remet à Ø après exécution.

Attention!! A chaque fois que l'ORIC rend la main et affiche Ready, l'écran est à nouveau sélectionné. En mode programme cela ne crée pas de problème, mais en mode direct il vous faudra utiliser plusieurs instructions par ligne. Par exemple : POKE # 2F1, 128 : LIST <RETURN> équivaut à LLIST (pourquoi faire simple, quand on peut faire de manière compliqué?). Si vous faites : POKE # 2F1, 128 <RETURN> LIST <RETURN> votre imprimante ne fera pas grand bruit, vous n'obtiendrez que... LIST.

Quant aux possesseurs de disquettes, ils utiliseront les instructions !PRINTER ON et !PRINTER OFF qui correspondent exactement à POKE #2F1, 128 et à POKE #2F1,Ø. Là encore, pas de problème en mode programme, mais en mode direct, il faudra mettre sur une même ligne : !PRINTER ON : !DIR par exemple.

Faute de savoir cela, certains s'imaginent que le matériel qu'ils possèdent est moins performant que celui du voisin. Voilà de quoi les rassurer.

Les possesseurs d'ATMOS, eux, sont mieux lotis : ils disposent de deux routines toutes prêtes qui réalisent ces fonctions en transférant aussi les longueurs des lignes.

CALL # C816 active l'imprimante.
CALL # C82F désactive l'imprimante, remet la sortie écran.

Ainsi pour l'ATMOS POKE # 2F1,128 est avantageusement remplaçable par CALL # C816 et POKE # 2F1,Ø par CALL # C82F.

EST-IL LONG, MON PROGRAMME ?

Si, pour des raisons tout à fait futiles, vous voulez savoir combien de lignes occupe votre programme, deux solutions s'offrent à vous :

1. - Les compter.
2. - Si vous possédez une **renuméroteur,** faire une renumérotation de 1 en 1 à partir de 1 et le numéro de la dernière ligne coïncidera avec le nombre de lignes (élémentaire, mon cher...).

Je vous propose la... troisième solution :

3. - Si N est la dernière ligne de votre programme, faites DOKE # 1D,Ø : GOTO N)ou N+1 si vous ne voulez pas qu'il démarre).
   Ignorez l'éventuel "UNDEF"D STATMENT.
   Écrivez alors PRINT DEEK (#1D) <RETURN>et le tour est joué.
   EXPLICATION : Lors de la recherche d'un n° de ligne, le compteur en # 1D est incrémenté de 1 à chaque ligne rencontrée. Il n'est remis à zéro que par un RUN ou l'entrée d'une nouvelle ligne.

LE PROGRAMME LE PLUS COURT

Les heureux possesseurs du lecteur de micro-disquettes ORIC auront remarqué le programme "OLD.COM", qui sert à récupérer les programmes BASIC. La méthode classique est : POKE # 5Ø2,X avec X≠Øet CALL # C56F (#C55F pour l'ATMOS). Ici, ce fichier est réduit à un seul octet, non nul (il vaut 5) qui va se placer en # 5Ø2. Quant au CALL, le DOS exécute toujours cette routine après le chargement d'un programme BASIC.

LES BOSSES DU DOGUE

Les disquettes sont arrivées. Quelle chance ! Nous allons pouvoir ouvrir la rubrique "Les bogues du D.O.S.".

Vous avez remarqué que l'initialisation faites par le D.O.S. est différente. En particulier, elle ne remplit pas la mémoire de "U". Cela, c'est pratique. Ce qui l'est moins c'est qu'elle oublie de positionner les variables systèmes régissant la longueur d'une ligne d'écran, ou plutôt, elle les positionne mal : elle met 8Ø en # 31...!

Il sera donc utile d'intégrer dans le "BOOT UP" les lignes suivantes :

ORIC 1 : POKE # 31,53
(vous vous souvenez ?).
ATMOS : POKE # 31,4Ø : POKE # 256,8Ø.

Certains inconvénients comme le saut ligne intempestif de l'imprimante sont ainsi supprimés.

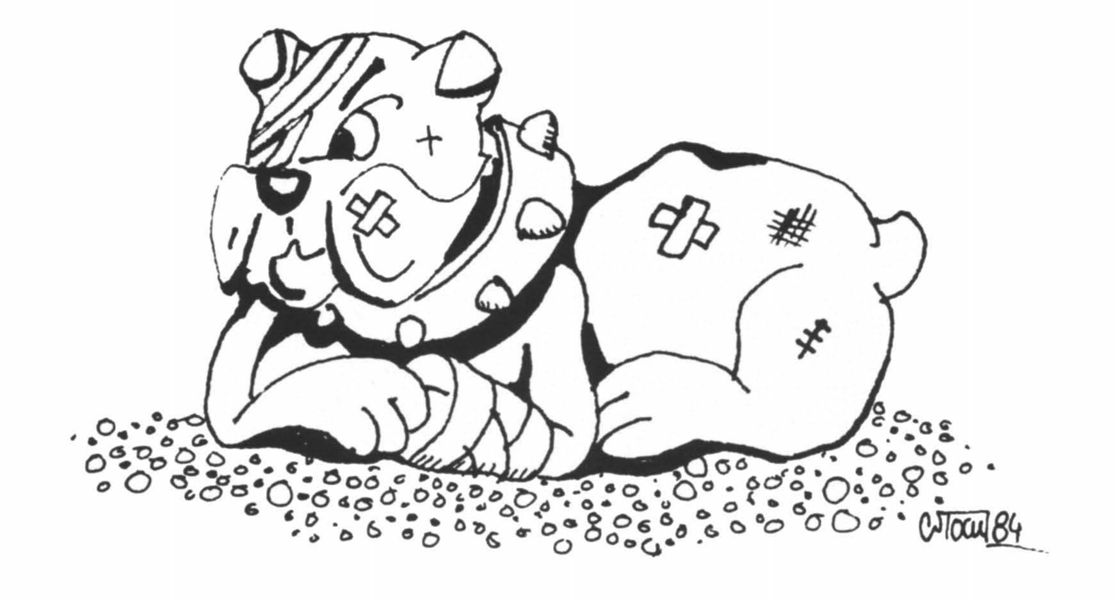
L'EXPLICATION

C'est maintenant bien connu, la tabulation horizontale sur ORIC 1 souffre d'un décalage de 13 caractères. En voici l'explication.

Il existe une routine en ROM qui envoie un retour chariot et un saut à la ligne. Elle est constamment appelée par PRINT. Son adresse est # CB9F. Voici son listing :

LDA 13	Charger A avec le code "retour chariot'.
STA #30	Mettre le contenu de A à l'adresse 30. **ERREUR ! Il faut mettre 0 et non 13 à cette adresse !**
JSR # CC12	Afficher "retour chariot".
LDA 1Ø	Charger A avec le code "curseur vers le bas".
JSR # CC12	Afficher "curseur vers le bas".

Un POKE # CBAØ,Ø ne sert à rien !



DISQUETTES ET LANGAGE MACHINE

Si vous avez essayé d'enregistrer, puis de lire des programmes en langage machine, vous aurez sans doute connu quelques déboires. C'est parce que l'ORIC est en position "RAM" quand le programme est chargé ; c'est-à-dire que la ROM BASIC est occultée et remplacée par la RAM contenant le DOS.

Donc, si votre programme utilise des routines de la ROM, il ira se perdre puisqu'il exécutera en fait des routines du DOS : GARE !!

Voici une méthode permettant d'éviter cet inconvénient : insérer dans votre programme un ordre autorisant à repasser sur la ROM, soit ;

```
LDA % # Ø2
JSR # 4E6
```

Un autre problème :

Si l'on sort du programme par un RTS, on perd la main ou l'on reçoit un SYNTAX ERROR, ceci étant du à une mauvaise gestion de la pile. Pour éviter cela, remplacer RTS par :

JMP #CØØ3 si vous êtes sur la ROM

ou JMP #D463 si vous êtes sur la RAM.



COUCOU C'EST MOI !

Vous possédez un ORIC 1, faites donc CALL #E7ØE. Appuyez sur une touche...Qui est le plus modeste? CTRL T Termine le jeu.

DESSIN PAR RECONFIGURATION DE CARACTÈRES

Les caractères reconfigurés sont dans la chaîne A$ en ligne 10. De la ligne 30 à la ligne 100 vous trouvez la procédure de reconfiguration

Pour chacun des 52 caractères, en ligne 40 on l'extrait de A$ pour le mettre en C$. En 50 on met son code ASC11 dans N. En 94 on prépare l'adresse de début. La boucle FOR.....NEXT en 95-98 exploite une ligne de DATA à chaque fois. Ainsi le signe [est reconfiguré par les "DATA" en ligne 107.

Ensuite on affiche 3 fois le dessin obtenu (boucle REPEAT....UNTIL). La boucle FOR....NEXT en 480-510 est destinée à l'animation du dessin. Le mouvement ayant lieu de gauche à droite, remarquez le caractère "espace" à la fin de chaque chaîne affichée par PLOT : il est indispensable pour provoquer l'effacement du dessin précédent.

JEEPS!

Ce dessin animé a été réalisé par 3 jeunes membres d'un club. (13-14 ans). Temps d'élaboration 4 à 5 heures.

```
1 CLS
10 A$="[\]_abcdefghijklmnopqrstuvwxyz{¦}
!#$%&'()*+,-./:;<=>"
30 FORJ=1TO52
40 C$=MID$(A$,J,1)
50 N=ASC(C$)
94 F=46080+(8*N)
95 FORI=FTOF+7
96 READB
97 POKEI,B
98 NEXTI
100 NEXTJ
107 DATA0,0,0,0,0,0,1,1
109 DATA7,12,15,15,15,15,63,15
110 DATA60,60,60,60,60,60,63,60
120 DATA31,51,63,63,63,63,63,63
130 DATA48,49,49,51,51,51,63,51
140 DATA0,60,14,62,62,62,63,62
150 DATA0,0,0,0,0,0,32,32
160 DATA0,0,0,0,0,1,2,4
170 DATA0,0,8,24,32,0,0,0
180 DATA3,3,2,2,4,5,5,9
190 DATA15,63,0,0,63,0,0,0
200 DATA60,63,2,2,58,10,10,10
210 DATA63,63,0,0,63,32,32,32
220 DATA51,63,0,0,63,1,1,1
230 DATA62,63,0,0,62,2,2,2
240 DATA48,48,16,24,28,21,21,22
250 DATA8,16,16,32,32,0,0,0
260 DATA0,0,0,0,0,0,0,63
270 DATA28,18,19,19,31,3,15,48
280 DATA10,10,10,18,20,20,55,8
290 DATA0,0,0,0,0,0,63,0
300 DATA10,10,10,10,10,10,58,2
310 DATA32,32,32,32,56,8,15,0
320 DATA1,1,1,1,1,1,63,0
330 DATA2,2,2,2,2,2,62,0
340 DATA18,18,18,18,18,19,31,3
350 DATA0,0,0,0,0,60,60,12
360 DATA7,52,52,60,52,52,52,52
370 DATA63,0,0,0,0,0,0,0
380 DATA8,15,8,4,4,4,7,4
390 DATA58,34,4,4,4,4,8,8
400 DATA3,3,3,3,3,3,3,3
410 DATA12,12,12,12,12,12,12,12
420 DATA60,52,52,55,60,52,52,6
430 DATA3,12,48,1,7,15,31,30
440 DATA60,2,49,56,62,63,63,7
450 DATA0,0,0,32,16,15,39,35
460 DATA2,2,1,0,0,48,63,63
461 DATA0,0,0,63,0,0,63,63
462 DATA8,8,16,32,0,0,63,63
463 DATA0,0,3,4,9,19,39,7
464 DATA30,33,12,30,63,63,63,33
465 DATA0,0,48,8,36,50,59,56
466 DATA3,3,3,2,2,2,62,62
467 DATA12,60,60,0,0,0,0,0
468 DATA62,62,62,31,31,15,3,0
469 DATA55,55,7,63,63,63,60,48
470 DATA33,49,49,33,32,0,0,0
471 DATA15,15,15,7,7,3,0,0
472 DATA45,45,33,63,63,63,63,12
473 DATA60,60,60,56,56,48,0,0
474 DATA48,16,16,0,0,0,0,0
475 REPEAT
476 K=K+8:Y=Y+5
480 FORX=27 TOKSTEP-1
485 WAIT20
500 PLOTX,2+Y,"[\]_abcde "
501 PLOTX,3+Y,"fghijklm "
502 PLOTX-2,4+Y,"nopqrstuvw "
503 PLOTX-4,5+Y,"xy  z {   ¦} "
504 PLOTX-4,6+Y,"!#$%&'()*+,- "
505 PLOTX-3,7+Y,"./:    ;<=> "
510 NEXT
520 UNTILK=24
```





HORLOGE

par Sylvain BRISSET

Voici un petit programme en langage machine pour ORIC-1. Pour l'écrire utiliser par exemple cette méthode, en relevant les nombres en hexadécimal dans le listing désassemblé.

Pour le sauver en langage machine sur cassette faire CSAVE "HORLOGE", A1024, E 1187.

Pour l'utilisation après chargement faire NEW.

La mise à l'heure se fait par :
DOKE 18,48015:PRINT "12:35:47" par exemple.

Le démarrage de la pendule est obtenu par :
DOKE 553,1157 ou CALL # 485

Si vous voulez de la couleur, il vous suffit de POKER en 48000 le nombre de 1 à 7 qui vous plaît.

Vous conservez l'heure à l'écran en écrivant votre programme, en l'exécutant mais avec certaines restrictions : ne pas atteindre la ligne d'état par CSAVE ou CLOAD. Ne pas passer en mode HIRES.

D'autre part, votre HORLOGE n'indiquera plus l'heure exacte si vous utilisez WAIT ou si vous modifiez la vitesse du clavier ou si vous inhibez les interruptions du clavier pour utiliser une imprimante par exemple.

Voici quelques commentaires du langage machine :

403 on teste si le chiffre des unités des secondes est à 9.
407 on ajoute 1 seconde.
412 on teste si le chiffre des dizaines des secondes est à 5.
416 on ajoute 1 au chiffre des dizaines de secondes.
421 on teste si le chiffre des unités des minutes est à 9.
425 on ajoute 1 minute.
430 on teste si le chiffre des dizaines des minutes est à 5.
434 on ajoute 1 au chiffre des dizaines de minutes.
43F on teste si le chiffre des unités des heures est à 9.
446 on teste si le chiffre des unités des heures est à 3.
44A on ajoute 1 au chiffre des unités des heures.
453 on teste si le chiffre des dizaines des heures est à 2.
457 on ajoute 1 au chiffre des unités des heures.
462 on ajoute 1 au chiffre des dizaines des heures.
46A-479 mise à zéro diverses.
482 retour au programme d'interruption.
485 Pour commander l'exécution de ce programme, c'est ici q'il faut commencer.
48D saut à # 481 si le TIMER n'indique pas 100 centièmes de secondes.
490 saut à # 4ØØ si le TIMER est arrivé à 100 centièmes de secondes.

```
0400 AD98BB LDA #BB98
0403 C939   CMP %#39
0405 F008   BEQ #040F
0407 EE98BB INC #BB98
040A A930   LDA %#30
040C 4C7C04 JMP #047C
040F AD97BB LDA #BB97
0412 C935   CMP %#35
0414 F008   BEQ #041E
0416 EE97BB INC #BB97
0419 A930   LDA %#30
041B 4C7904 JMP #0479
041E AD95BB LDA #BB95
0421 C939   CMP %#39
0423 F008   BEQ #042D
0425 EE95BB INC #BB95
0428 A930   LDA %#30
042A 4C7604 JMP #0476
042D AD94BB LDA #BB94
0430 C935   CMP %#35
0432 F008   BEQ #043C
0434 EE94BB INC #BB94
0437 A930   LDA %#30
0439 4C7304 JMP #0473
043C AD92BB LDA #BB92
043F C939   CMP %#39
0441 F01F   BEQ #0462
0443 AD92BB LDA #BB92
0446 C933   CMP %#33
0448 F006   BEQ #0450
044A EE92DD INC #DD92
044D 4C9D04 JMP #049D
0450 AD91BB LDA #BB91
0453 C932   CMP %#32
0455 F008   BEQ #045F
0457 EE92BB INC #BB92
045A A930   LDA %#30
045C 4C9804 JMP #0498
045F 4C9304 JMP #0493
0462 EE91BB INC #BB91
0465 A930   LDA %#30
0467 4C6D04 JMP #046D
046A 8D91BB STA #BB91
046D 8D92BB STA #BB92
0470 8D94BB STA #BB94
0473 8D95BB STA #BB95
0476 8D97BB STA #BB97
0479 8D98BB STA #BB98
047C A9FF   LDA %#FF
047E 8D7602 STA #0276
0481 68     PLA
0482 4C03EC JMP #EC03
0485 48     PHA
0486 AD7602 LDA #0276
0489 C99B   CMP %#9B
048B F003   BEQ #0490
048D 4C8104 JMP #0481
0490 4C0004 JMP #0400
0493 A930   LDA %#30
0495 4C6A04 JMP #046A
0498 A930   LDA %#30
049A 4C7004 JMP #0470
049D A930   LDA %#30
049F 4C7004 JMP #0470
10 I=1024
20 REPEAT
30 READA$
40 A=VAL(A$)
50 POKEI,A
60 I=I+1
70 UNTILA$="FIN"
100 DATA#AD,#98,#BB,#C9....
300 DATA....#4C,#70,#04,FIN
```



DES CHIFFRES OU DES LETTRES ?

par Thierry TOSELLO

Voici pour ORIC-1 ou ATMOS un programme classique de conversion d'un nombre entier écrit en chiffres en son expression en toutes lettres. C'est le problème qui se pose à une banque qui veut éditer des chèques automatiquement.

Ce programme est perfectible, en particulier en y ajoutant le traitement de la partie décimale. La méthode employée consiste à découper le nombre en tranches de trois chiffres.

Commentaire du listing :

Ligne 8 :
Choisissez les couleurs que vous voulez.
?CHR$(17) pour enlever le curseur.
HIMEM #97FF pour ORIC-1.

Lignes 9-10 :
Sur ATMOS vous pouvez utiliser PRINT AT.

Lignes 25-26 :
Contrôle du nombre introduit.

Lignes 27-37 :
Mise en forme et affichage par tranche de 3 chiffres. Préparation du drapeau FL pour le "S" à CENT.

Lignes 180-220 :
Aiguillage principal.

Lignes 250-... :
Les milliards.

Lignes 300-... :
Les millions.

Lignes 350-... :
Les mille.

Lignes 400-... :
Les unités.

Lignes 54-100 :
Aiguillages successifs pour former les mots.

Lignes 900-910 :
Affichage du résultat. Le "A" en ligne 9ØØ commande le rouge. Changer la lettre pour avoir une autre encre.
Remarquez l'usage de POS(Ø) pour contrôler la fin des lignes.

Pour ATMOS il est préférable d'écrire :

```
900 PRINT:PRINT:PRINT:PRINTCHR$(27)+"A"
;
908 IFPOS(0)<3THENPRINT:PRINTCHR$(27)+"
A";
```

```
8 PAPER6:INK4:PRINTCHR$(17):HIMEM#97FF
9 CLS:PRINT:PRINT:PRINT:PRINT
10 PRINT"Donnez un nombre inferieur a :"
:PRINT
12 PRINT"           1000 milliards"
14 U$="000000000000"
21 DIMA$(255)
25 PRINT:PRINT:PRINT:INPUT P$:P=VAL(P$):
IFP<0ORINT(P)<>PTHENPING:CLEAR:GOTO9
26 IFLEN(P$)>12THENZAP:CLEAR:GOTO9
27 P$=U$+P$
30 P$=RIGHT$(P$,12):L=12
32 IF VAL(LEFT$(P$,3))=0THENP$=RIGHT$(P$
,L-3):L=L-3:IFL<>0GOTO32
33 IFL=0THENP$="000":L=3
35 W$=RIGHT$(P$,3):F$=LEFT$(W$,1):F=VAL(
F$):G$=RIGHT$(W$,2):G=VAL(G$)
36 IFF>1ANDG=0THENFL=1ELSEFL=0
37 CLS:PRINT:PRINTSPC((35-LEN(P$))/2);:F
ORU=1TOLSTEP3:PRINTMID$(P$,U,3)" ";:NEXT
```

```
40 GOTO 160
45 REM
50 REM   Sous-programme principal
52 REM
54 MG=VAL(MID$(N$,1,1))
55 M=VAL(MID$(N$,2,1))
56 T=VAL(RIGHT$(N$,1))
57 IF MG=0 THEN 72
58 IF MG=1 THEN 61
60 ON MG  GOSUB 1100,1200,1300,1400,1500
,1600,1700,1800,1900
61 Z$=Z$+" CENT "
62 Z$=Z$+" "
72 IF M=0 THEN GOTO 97
75 IF M=1 THEN 81 ELSE 90
81 IF T=0THEN Z$=Z$+"DIX":RETURN
82 ON T GOSUB 2100,2200,2300,2400,2500,2
600,2700,2800,2900
85 RETURN
90 ON M GOSUB 3100,3200,3300,3400,3500,3
600,3600,3800,3800:Z$=Z$+" "
91 IF M=7 OR M=9 THEN 92 ELSE 95
92 IF T=1 THEN Z$=Z$+" ET "
93 Z$=Z$+" ":IF T=0THEN Z$=Z$+"DIX "
94 ON T GOSUB 2100,2200,2300,2400,2500,2
600,2700,2800,2900:RETURN
95 IF T=1 THEN Z$=Z$+" ET "
97 ON T GOSUB 1100,1200,1300,1400,1500,1
600,1700,1800,1900
100 RETURN
150 REM
160 REM Programme principal
170 REM
180 IF P$="000" THEN Z$="ZERO":GOTO900
200 IF L>9 THEN 250
210 IF L>6 THEN 300
220 IF L>3 THEN 350 ELSE 400
250 N$=LEFT$(P$,3)
260 GOSUB 50:Z$=Z$+" MILLIARD"
265 IF VAL(N$)>=2THEN Z$=Z$+"S ":GOTO 50
0 ELSE Z$=Z$+" ":GOTO 500
300 N$=LEFT$(P$,3)
305 IF N$="000" THEN 500
310 GOSUB 50:IF Z$="000" THEN 500
315 Z$=Z$+" MILLION"
320 IF VAL(N$)>=2 THEN Z$=Z$+"S ":GOTO 5
00 ELSE Z$=Z$+" ":GOTO 500
350 N$=LEFT$(P$,3)
360 GOSUB 50:IF N$="000" THEN 500
370 IFRIGHT$(Z$,2)="UN"THENZ$=LEFT$(Z$,L
EN(Z$)-2)
375 Z$=Z$+" MILLE ":GOTO 500
400 N$=LEFT$(P$,3)
450 GOSUB 50
500 IF L<4 THEN 900
800 P$=RIGHT$(P$,L-3)
820 L=LEN(P$)
850 GOTO 200
900 PRINT:PRINT:PRINT:PRINT"  "+CHR$(27)+
"A";
901 IFRIGHT$(Z$,1)=" "THENZ$=LEFT$(Z$,LE
N(Z$)-1):GOTO901
904 IFFL=1THENZ$=Z$+"S"
905 FORI=1TOLEN(Z$):A$(I)=MID$(Z$,I,1):I
FA$(I)<>" "THENPRINTA$(I);
906 IFA$(I)=" "ANDA$(I-1)<>" "THENPRINTA
$(I);
907 IFPOS(0)>31 AND(ASC(A$(I))=32 OR ASC
(A$(I))=45)THENPRINT
908 IFPOS(0)<2THENPRINT:PRINT"  "+CHR$(27
)+"A";
909 NEXTI:PRINT"."
910 CLEAR
920 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
950 PRINT "Un autre ?   (O/N)":GET C$
960 IF C$="O" THEN 9 ELSE END
980 REM
990 REM Sous-programme
1000 REM  de donnees
1050 REM
1100 Z$=Z$+"UN":RETURN
1200 Z$=Z$+"DEUX":RETURN
1300 Z$=Z$+"TROIS":RETURN
1400 Z$=Z$+"QUATRE":RETURN
1500 Z$=Z$+"CINQ":RETURN
1600 Z$=Z$+"SIX":RETURN
1700 Z$=Z$+"SEPT":RETURN
1800 Z$=Z$+"HUIT":RETURN
1900 Z$=Z$+"NEUF":RETURN
2100 IFRIGHT$(Z$,4)="ET  "THENZ$=LEFT$(Z
$,LEN(Z$)-4)+" "
2110 Z$=Z$+"ONZE":RETURN
2200 Z$=Z$+"DOUZE":RETURN
2300 Z$=Z$+"TREIZE":RETURN
2400 Z$=Z$+"QUATORZE":RETURN
2500 Z$=Z$+"QUINZE":RETURN
2600 Z$=Z$+"SEIZE":RETURN
2700 Z$=Z$+"DIX-SEPT":RETURN
2800 Z$=Z$+"DIX-HUIT":RETURN
2900 Z$=Z$+"DIX-NEUF":RETURN
3200 Z$=Z$+"VINGT":RETURN
3300 Z$=Z$+"TRENTE":RETURN
3333 IFL=0THENP$="0"
3400 Z$=Z$+"QUARANTE":RETURN
3500 Z$=Z$+"CINQUANTE":RETURN
3600 Z$=Z$+"SOIXANTE":RETURN
3700 Z$=Z$+"SOIXANTE-DIX":RETURN
3800 Z$=Z$+"QUATRE-VINGT":RETURN
3900 Z$=Z$+"QUATRE-VINGT-DIX":RETURN
```



Voici un programme destiné à recopier l'écran HIRES sur l'imprimante SEIKOSHA GP50A. D'abord en BASIC.

par Georges BARRET

BASIC

```
5 CALL#F960:POKE49,58:LPRINTCHR$(27)"0"
10 FORI=0TO199STEP8:LPRINTCHR$(27)"G"CHR$(0)C
HR$(#E4);
20 FORJ=12TO239:K=0:FORL=0TO7:IFPOINT(J,I+L)=
-1THENK=K+2^L
30 NEXTL:LPRINTCHR$(K);:NEXTJ:LPRINTCHR$(0):N
EXTI
40 CALL#E804
50 PING:WAIT20:GOTO50

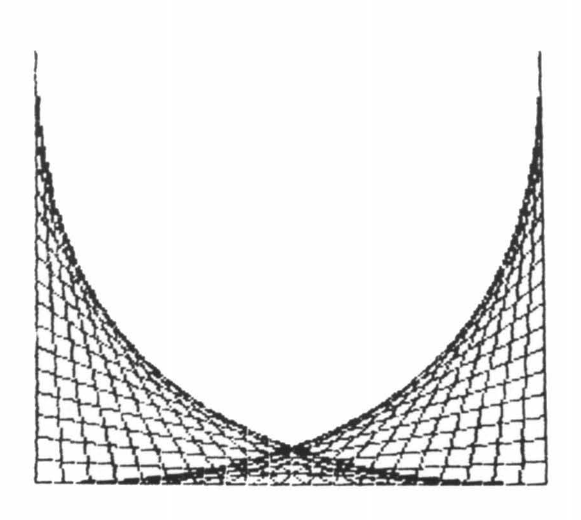
c'est tres long !!!!
        Georges BARRET
```

Puis en langage machine, appelé par CALL 17500.

Ce programme utilise une astuce qui le rend inutilisable sur la GP100. Sachant que la GP50 imprime en graphique, 8 points alignés et que l'ORIC dispose, en correspondance de colonnes de 6 points, on peut facilement avec 4 colonnes de 6 points faire 3 rangées de 8 points. Cette remarque est à la base de la conception de ce programme.

LANGAGE MACHINE

```
10 REM ===== HIRES'COPY =====
20 REM == for GP 50 A,only ==
30 REM
40 REM (a la memoire d'un rouleau de papier
50 REM que j'ai beaucoup aime )
60 REM
70 REM  (c)  Georges BARRET - 30 mars 84
80 REM
1000 FORI=#7500TO#768F
1010 READ P$:P=VAL("#"+P$)
1020 POKEI,P
1030 NEXT
2000 DATA 20,83,76,EA,EA,A9,0C,20,12,CC,A9,1B
,20,7B,F5,A9,30,20,7B,F5,4C,5E
2010 DATA 75,00,C9,20,10,02,A9,00,C9,80,30,02
,E9,80,C9,40,30,02,E9,40,18,60
2020 DATA 98,48,A9,1B,20,7B,F5,A9,47,20,7B,F5
,A9,00,20,7B,F5,A9,C8,20,7B,F5
2030 DATA 68,A8,60,98,48,A9,0A,20,7B,F5,68,A8
,60,98,48,8A,48,AD,17,75,20,7B
2035 DATA F5,68,AA
2040 DATA 68,A8,60,A0,28,20,2C,75,A2,00,B9,FF
,9F,20,18,75,8D,17,75,B9,FE,9F
2050 DATA 20,18,75,29,03,0A,0A,0A,0A,0A,0A,6D
,17,75,8D,17,75,20,4F,75,AD,66
2060 DATA 75,69,28,90,04,EE,67,75,18,8D,66,75
,AD,6F,75,69,28,90,04,EE,70,75
2070 DATA 18,8D,6F,75,E8,E0,C8,D0,BF,A9,9F,8D
,67,75,8D,70,75,A9,FF,8D,66,75
2080 DATA A9,FE,8D,6F,75,20,45,75
2085 DATA20,2C,75,A2,00,B9,FE,9F,20,18,75,29,
3C,4A,4A,8D,17
2090 DATA 75,B9,FD,9F,20,18,75,29,0F,0A,0A,0A
,0A,6D,17,75,8D,17,75,20,4F,75
2100 DATA AD,C1,75,69,28,90,04,EE,C2,75,18,8D
,C1,75,AD,CE,75,69,28,90,04,EE
2110 DATA CF,75,18,8D,CE,75,E8,E0,C8,D0,BD,A9
,9F,8D,C2,75,8D,CF,75,A9,FE,8D
2120 DATA C1,75,A9,FD,8D,CE,75,20,45,75,20,2C
,75,A2,00,B9,FD,9F,20,18,75,29
2130 DATA 30,4A,4A,4A,4A,8D,17,75,B9,FC,9F,20
,18,75,29,3F,0A,0A,6D,17,75,8D
2140 DATA 17,75,20,4F,75,AD,1E,76,69,28,90,04
,EE,1F,76,18,8D,1E,76,AD,2D,76
2150 DATA 69,28,90,04,EE,2E,76,18,8D,2D,76,E8
,E0,C8,D0,BD,A9,9F,8D,1F,76,8D
2160 DATA 2E,76,A9,FD,8D,1E,76,A9,FC,8D,2D,76
,20,45,75,88,88,88,88,C0,00,F0
2170 DATA 03,4C,60,75,4C,8C,76,20,60,F9,A9,11
,20,12,CC,60,20,04,E8,60
```



Nous remercions M. Georges BARRET, nous publierons des programmes pour diverses imprimantes.

Monsieur Georges BARRET - 12 Baraqueville...

Vous propose un programme d'affichage en **lettres géantes** qui défilent, très utile pour des présentations, des démonstrations. Vous pouvez entrer jusqu'à 11 lignes de 39 caractères, choisir les couleurs, la vitesse de défilement.

Pour mettre une virgule, affichez £ sinon... essayez la vraie ! @ a été remplacé par un carré noir, c'est utile.

Ce programme est prévu pour ORIC-1, les possesseurs d'ATMOS feront les adaptations connues. POKE 524,127 fait passer en minuscules sans enlever CAPS. POKE 524,255 fait revenir en majuscules.




```
1 REM ****  BANDEROLLE  ****
2 REM --- Georges BARRET ---
3 REM
5 HIMEM#97FF:TEXT:POKE618,10:POKE524,127
7 FORI=48000TO48039:POKEI,32:NEXTI
10 FORI=46592TO46599:POKEI,63:NEXTI
15 AF$="0000000000080816":FORI=1TO8:POKE46839+I,VAL(MID$(AF$,I*2-1,2)):NEXTI
25 CLS:PAPER0:INK3:PRINT:PRINT:PRINT
30 INPUT"COULEUR D'ENCRE (de 0 a 7): ";EN:PRINT:PRINT
31 PRINT"(0) FOND NOIR ":PRINT:INPUT"(1) FOND INVERSE DE L'ENCRE ";R$:PRINT:PRIN
T
32 IFR$="0"THENFO=0 ELSEFO=7-EN
33 PRINT"VOTRE MESSAGE.....":PRINT"11 lignes de 39 caracteres, maximum."
34 PRINT:INPUTMO$(M):IFLEN(MO$(M))>39THENEXPLODE:PRINT:PRINT"TROP LONG!":PRINT:G
OTO34
37 MO$(M)=MO$(M)+CHR$(32)+CHR$(32¨+CHR$(32):IFMO$(M)="FIN   "THEN40
39 IFM<10THENM=M+1:GOTO34ELSE44
40 MO$(M)="":M=M-1
44 PRINT:PRINT:INPUT"NOMBRE DE REPETITIONS: ";NR
45 PRINT:PRINT:PRINT"VITESSE DE DEFILEMENT:":PRINT"de 1 (lent) a 8 (rapide)"
50 GETR$:V=VAL(R$):IFV<1ORV>8THEN45
51 IFV=5THENV=6
52 IFV=6THENV=9
53 IFV=7THENV=12
54 IFV=8THENV=18
65 CLS:CALL#E6CA
66 FORI=13TO14:PLOT2,I,14:PLOT5,I,"JE TRAVAILLE...":NEXTI
70 FORMM=0TOM:FORI=1TOLEN(MO$(MM)):A$=MID$(MO$(MM),I,1):FORJ=0TO7
75 N=46080+ASC(A$)*8+J:X=PEEK(N):V$="":REPEAT:Q=INT(X/2):R=X-Q*2
85 IFR=1THENR$=CHR$(64)ELSER$=CHR$(32)
90 V$=R$+V$:X=Q:UNTILX=0
100 IFLEN(V$)<6THENV$=CHR$(32)+V$:GOTO100
102 LI$(J,MM)=LI$(J,MM)+V$:NEXTJ,I,MM:CALL#E804
136 CLS:FORI=13TO14:PLOT2,I,10:PLOT9,I,"PRESSEZ UNE TOUCHE":NEXTI
137 FORI=17TO18:PLOT2,I,10:PLOT8,I,"POUR LANCER LE TEXTE":NEXTI
138 PING:FORT=1TO500:K$=KEY$:IFK$<>""THEN145
139 NEXTT:GOTO138
145 PAPERFO:CLS:CALL#E6CA:PLOT1,25,EN:PLOT10,25,CHR$(96)+" G. Barret 1983"
150 FORZ=1TONR:FORMM=0TOM:FORK=37TO1STEP-V
160 I¨KFO:FORJ=0TO7:PLOTK,10+J,MID$(LI$(J,MM),1,39-K):NEXTJ:INKEN
210 FORW=1TO150:NEXTW,K:FORI=VTOLEN(LI$(0,MM))STEPV
240 INKFO:FORJ=0TO7:PLOT1¨10+J,MID$(LI$(J,MM),I,38):NEXTJ:INKEN
270 FORW=1TO150:NEXTW,I,MM,Z:CALL#E804
```

Voici un programme proposé par M. WAUQUIER qui permet la copie d'écran HIRES sur MCP40

Sur ORIC-1 il faudra ajouter :

```
120 POKE 49,93 : DOKE 27,#E807
140 CALL #ED01
```

pour éviter les erreurs de transmission.

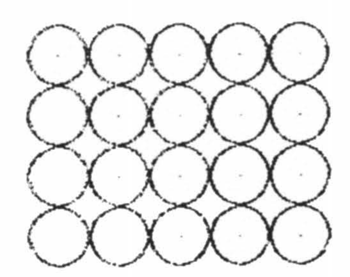
Les cercles déformés à l'écran, du fait de la forme rectangulaire des pixels, sont bien circulaires sur lè papier.

Pour doubler la taille, il faut modifier les lignes 270 à 310 en mettant "J2,0", "R2,0" et "M0,-2" et éventuellement avec "M0,-1" passer 2 fois dans la boucle N.

L'inconvénient d'un tel programme est son extrême lenteur. Il faut environ 50 s par ligne. Pour un écran complet c'est 2 h 45 mn.

Néanmoins il intéressera certains de nos lecteurs et nous pourrons proposer la version en langage machine ultérieurement.





```
10 REM DESSIN HAUTE RESOLUTION
20 REM
30 REM COPIE D'ECRAN HIRES
40 REM
50 REM SUR IMPRIMANTE MCP 40
55 REM
60 GOSUB 500
130 LPRINTCHR$(17)
150 LPRINTCHR$(18)
170 REPEAT
180 FOR X=40962+A TO 40999+A
200 B=PEEK(X)
220 FOR N=8 TO 1 STEP-1
230 A(N)=INT(B/2^(N-1)+.00001)
240 B=B-A(N)*2^(N-1)
250 NEXT N
260 FOR N=6 TO 1 STEP-1
270 IF A(N)=1 THEN LPRINT"J1,0"
280 IF A(N)=0 THEN LPRINT"R1,0"
290 NEXT N,X
310 LPRINT"M0,-1":LPRINT"I"
320 A=A+40
330 UNTIL A=7960
350 END
500 HIRES
510 FOR X=42 TO 202 STEP 40
520 FOR Y=30 TO 150 STEP 40
530 CURSETX,Y,1
540 CIRCLE20,1
550 NEXT Y,X
560 PRINT"PREPARER L'IMPRIMANTE"
570 PRINT:PRINT"APPUYER SUR UNE TOUCHE"
580 GETA$
590 RETURN
```

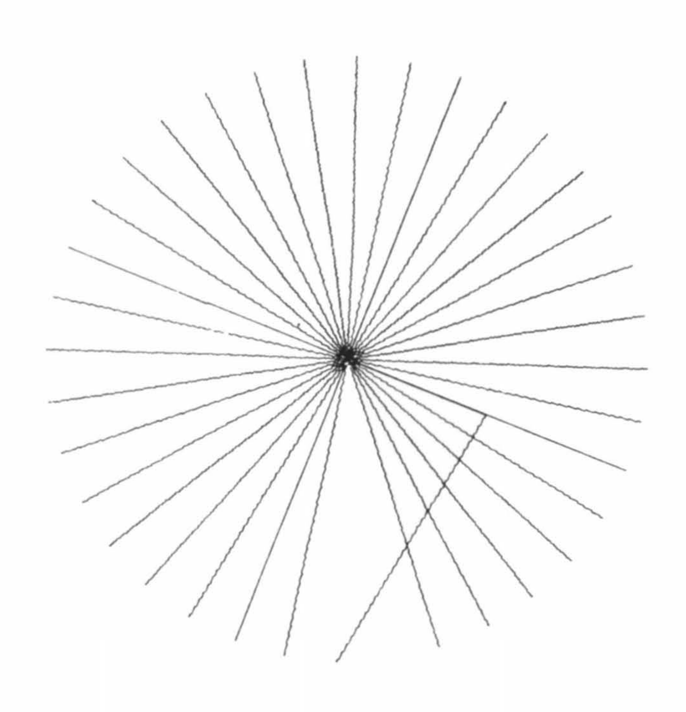
VARIANTE

```
270 IF A(N)=1 THEN LPRINT"J2,0"
280 IF A(N)=0 THEN LPRINT"R2,0"
290 NEXT N,X
310 LPRINT"M0,-2":LPRINT"I"
```

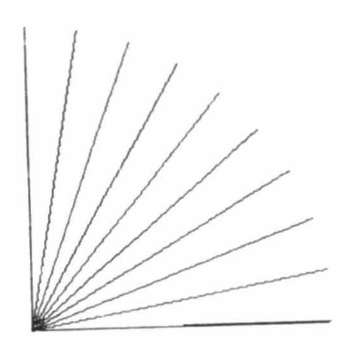
MCP 40 en mode graphique

Vous avez des ennuis avec la MCP 40 branchée à un ORIC-1 en mode graphique.

Voici des exemples de désagréments :



```
460 LPRINT CHR$(18);"L0"
470 LPRINT "M250,-180":LPRINT"I"
480 FOR I=0 TO 350 STEP 10
490 S=I/180*PI
500 X=SIN(S)*200.5:Y=COS(S)*200.5
510 X=INT(X):Y=INT(Y)
513 X$=CHR$(-32*(X>=0)-45*(X<0))+MID$(ST
R$(X),2)
517 Y$=CHR$(-32*(Y>=0)-45*(Y<0))+MID$(ST
R$(Y),2)
520 LPRINT "D";X$;",";Y$:LPRINT"H"
530 NEXT I
540 LPRINT"A"
```



```
10 LPRINT CHR$(18)
20 LPRINT"M250,-180":LPRINT"I"
30 FOR I=0 TO 350 STEP 10
40 S=I/180*PI
50 X=SIN(S)*200.5:Y=COS(S)*200.5
60 X=INT(X):Y=INT(Y)
70 LPRINT"D";X;",";Y:LPRINT"H"
80 NEXT I
90 LPRINT"A"
```

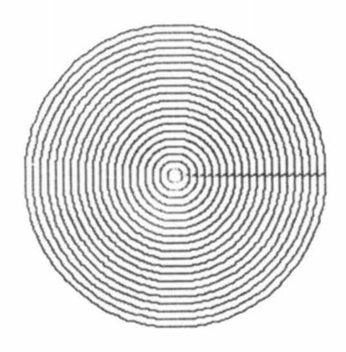
Le signe ☒ qui apparaît dans les textes est dû on le sait aux interférences entre la sortie imprimante et la lecture du clavier. En mode graphique la perturbation affecte les tracés.

Remède :

CALL # EDØ1 : LLIST ou CALL #EDØ1 : RUN

Un dessin qui devrait se développer sur 360° reste dans un angle de 90° (la zone correspondant à X négatif et Y négatif).

Ceci est dû au fait que sur ORIC-1 la fonction STR$ présente l'anomalie de coder # 2 au lieu de # 2Ø l'espace avant les chiffres. Il faut donc employer la parade connue (voir lignes 7Ø et 8Ø du programme dessinant des cercles concentriques.



CERCLES CONCENTRIQUES

```
10 LPRINTCHR$(18)

20 LPRINT"M200,-200"

30 LPRINT"I"

35 FORA=5TO100STEP5

40 FORI=0TOPI*2STEP.0314

50 X=COS(I)*A:X=INT(X)

60 Y=SIN(I)*A:Y=INT(Y)

70 X$=CHR$ (-32*(X>=0)-45*(X<0))+MID$(S
TR$(X),2)

80 Y$=CHR$ (-32*(Y>=0)-45*(Y<0))+MID$(S
TR$(Y),2)

90 LPRINT"D";X$;",";Y$

120 NEXTI

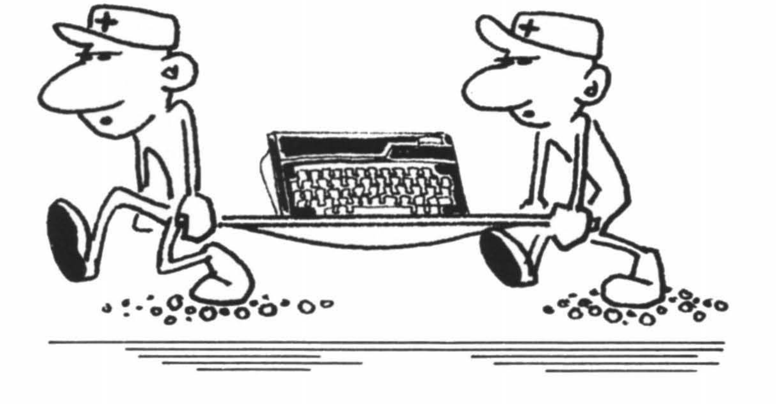
125 NEXTA

130 LPRINT"H"

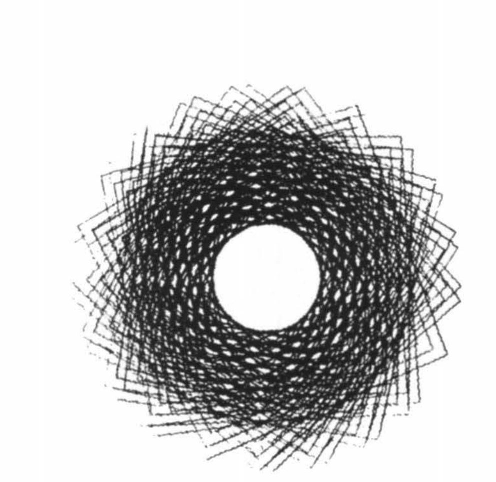
140 LPRINT"A"
```

GRAPHIQUES SUR MCP40

proposition de M. Thierry TOSELLO

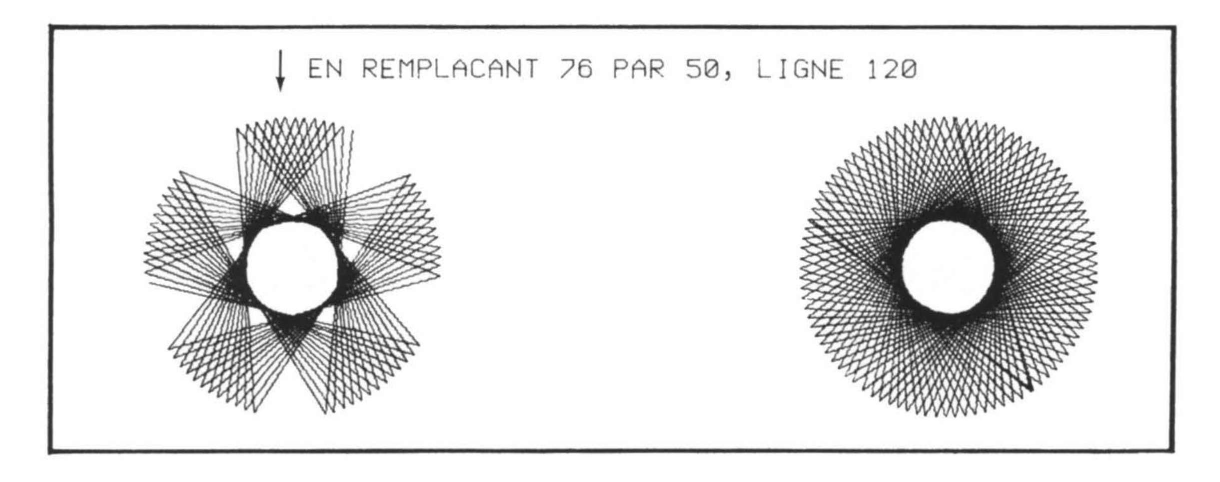


```
25 POKE #305,255
30 DEF FNF(X)=COS(X)*K
40 DEF FNG(X)=SIN(X)*K
50 K=100
100 LPRINT CHR$(18)
110 LPRINT "M240,0":LPRINT "I"
115 FOR K=50 TO 130 STEP 10
120 FOR T=0 TO 19*PI STEP 1.5
125 Z=INT(FNF(T)*10)/10
130 GOSUB 1000:X$=Z$
135 Z=INT(FNG(T)*10)/10
140 GOSUB 1000:Y$=Z$
150 IF T=0 THEN LPRINT "M";X$;",";Y$:GOT
O 200
160 LPRINT "D";X$;",";Y$
200 NEXTT,K
300 LPRINT "A":POKE #305,20:END
1000 Z$=CHR$(-32*(Z>=0)-45*(Z<0))+MID$(S
TR$(Z),2):RETURN
```



```
AUTRE MOTIF

EFFACER LA LIGNE 115
CHANGER LES LIGNES 120 ET 200
120 FOR T=0 TO 76*PI STEP 2.5
200 NEXT T
```



Vite fait Bien fait

NOMBRES PREMIERS

Les programmes permettant de calculer et afficher la liste des nombres premiers jusqu'à 1000 par exemple utilisent des méthodes variées.

Le temps est très variable. Le programme BASIC que voici met 12,5 secondes pour calculer et afficher les nombres premiers jusqu'à 1000 ! En ligne 2Ø et 15Ø vous trouverez une méthode de chronométrage qui pourrait servir ailleurs.

Idée de E. HOLLISTER publiée dans ORIC OWNER n° 6.

```
0 REM NOMBRES PREMIERS
1 REM
2 REM E. HOLLISTER
5 POKE618,10
10 CLS
20 DOKE630,65535
30 B=1000
40 DIM N(B)
50 A=SQR(B)
60 Z=INT(A)
70 FORX=3TOZSTEP2
80 FORW=X*3TOBSTEPX*2
90 N(W)=1
100 NEXT
105 NEXT
110 FORA=3TOB-1STEP2
120 IFN(A)=0THENPRINTA;
130 NEXT
140 PRINT:PRINT
150 PRINT(65535-DEEK(630
))/100" secondes ":PRINT
```

PGCD & PPCM

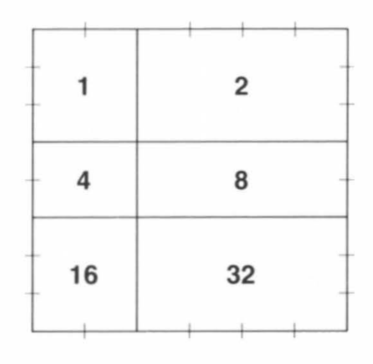
L'algorithme d'Euclide permet une programmation très facile pour la recherche des PGCD et PPCM.

Cette méthode est très connue, nous la fournissons pour ceux d'entre vous qui ne l'auraient pas encore rencontrée.

```
1 REM P.G.C.D. & P.P.C.M.
2 REM
10 CLS:PAPER0:INK6:POKE618,3
20 INPUT"ENTRER DEUX ENTIERS SVP ";A,B
21 PRINT:PRINT:POKE618,10
25 A1=A:B1=B
30 IFA=0ORB=0THENPRINT" PAS ZERO SVP ":W
AIT300:PING:GOTO10
40 Q=INT(A/B)
50 R=A-B*Q
60 IFR=0THEN100
70 A=B:B=R
80 GOTO40
100 PRINT"LE P.G.C.D DE "A1" ET "B1" EST
 "B
110 PRINT:PRINT:PRINT"LEUR P.P.C.M. EST
"A1*B1/B
120 GETA$:GOTO10
```

LORES 1

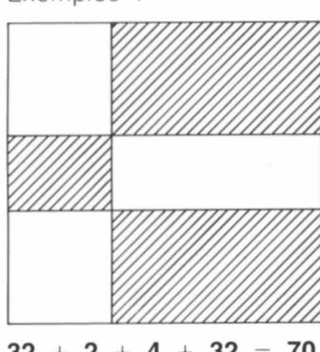
Les caractères semi-graphiques de l'ORIC ont une configuration facile à déduire de leur code ASCII. Ils sont dessinés à partir d'une grille de 6 cases.



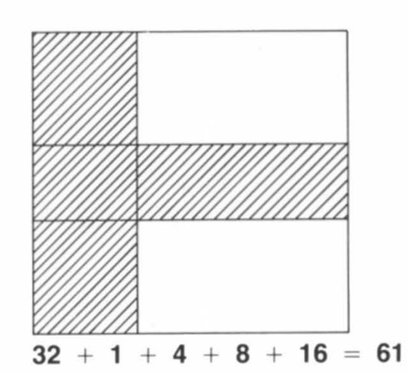
en hauteur **3 lignes**
2 lignes
3 lignes
en largeur **2 colonnes**
4 colonnes

Pour obtenir le code, on ajoute 32 à la somme des nombres figurant dans les cases.

Exemples :



32 + **2** + **4** + **32** = **70**



32 + **1** + **4** + **8** + **16** = **61**

Voici un moyen d'affichage.

```
5 CLS
10 LORES1
20 FORI=32TO95
30 PRINTCHR$(I)SPC(6);
40 NEXT
```

On pourrait souhaiter que les 2 colonnes aient la même largeur : voici un moyen de modifier les caractères.

```
60 FORI=47368TO47858
70 IFPEEK(I)=240THENPOKEI,56
80 IFPEEK(I)=15THENPOKEI,7
90 NEXT
```

Ceci devrait vous faciliter la création de dessins en page LORES 1.



Comment battre un jeu de cartes

Divers procédés sont envisageables.

Par exemple tirer 2 cartes au hasard et les échanger.

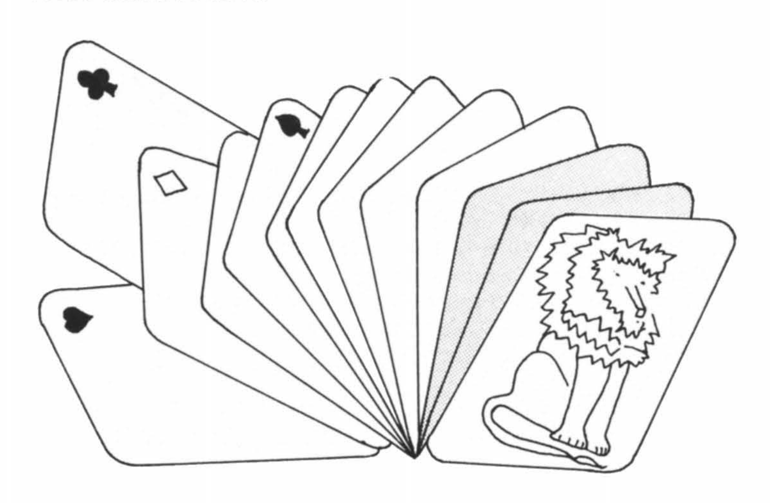
Répéter la même opération un certain nombre de fois.

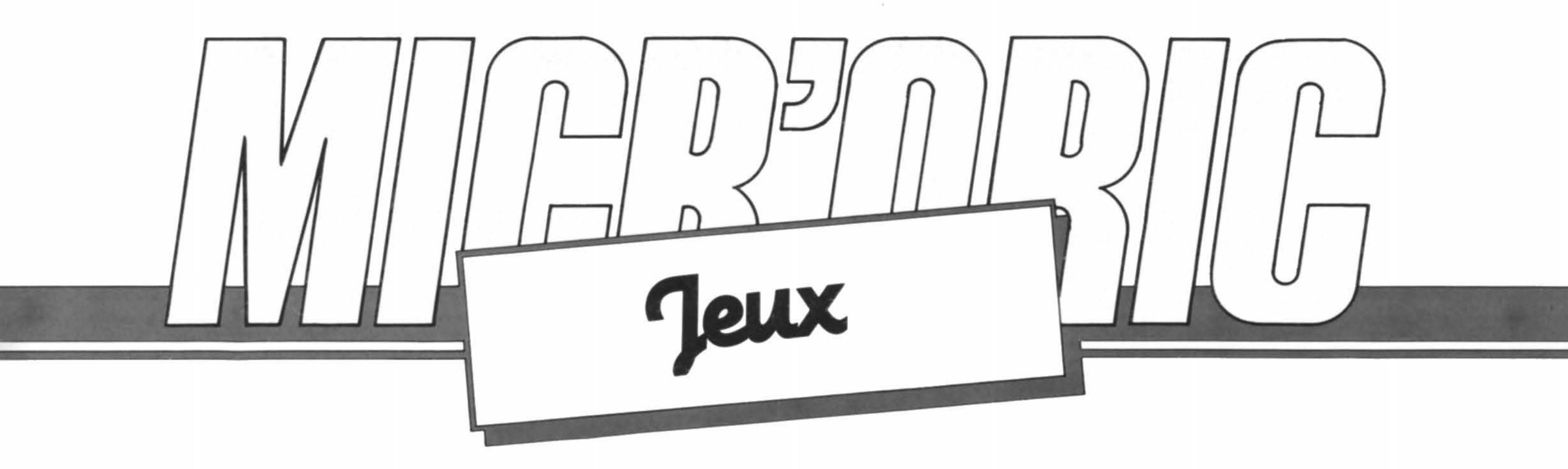
Une autre façon consiste à échanger chacune des cartes avec une autre de rang tiré au hasard.

Nous vous proposons une méthode de mélange qui peut servir en d'autres occasions pour un nombre plus élevés d'articles.

L'idée est de modifier en permanence la fourchette du tirage. Ainsi pour un jeu de 52 cartes. On tire une carte au hasard et elle devient la 52e. On tire dans les 51 premières une autre carte au hasard qui devient la 51e, etc... Toutes les cartes ont été affectées en un passage. Le résultat s'affiche sur votre écran en 4 mains de 13 cartes.

```
500 REM DONNE AU BRIDGE
502 CLS
505 HIMEM#97FF
510 DIMC$(52):DIMB$(13)
520 DATA" de trefle"," de carreau"," de
coeur"," de pique"
530 DATA"   As","    2","    3","    4",
"    5","    6","    7"
535 DATA"    8","    9","   10","valet",
" dame","  roi"
540 FORI=0TO3:READA$(I):NEXTI
550 FORJ=1TO13:READB$(J):NEXTJ
560 FORI=0TO3:FORJ=1TO13
570 C$(J+13*I)=B$(J)+A$(I)
580 NEXTJ,I
590 FORN=52TO1STEP-1
600 HZ=INT(RND(1)*N+1)
610 X$=C$(HZ):C$(HZ)=C$(N):C$(N)=X$
620 NEXTN
625 FORJ=0TO3
627 PRINT:PRINT"MAIN No "J+1:PRINT:PRINT
630 FORI=1TO13:PRINT C$(I+J*13):NEXTI:GE
TA$:CLS:NEXTJ
```





DOLLAR MAN

par Christophe ANDRÉANI

Ce jeu est entièrement programmé en BASIC pour ORIC 1. Pour l'adopter à l'ATMOS c'est facile et nous l'indiquons à la fin. Pour gagner de la vitesse on inhibe la scrutation du clavier.

Le but du jeu est expliqué dans le programme. Les règles en sont simples mais pertinentes.

Le personnage que vous guidez avec les touches "Q" et "E" dans ses déplacements horizontaux et "P" et "L" dans ses déplacements verticaux fait entendre son essouflement lorsqu'il s'arrête.

La difficulté augmente avec le nombre de niveaux.

....

Voici quelques indications pour adaptation à l'ATMOS :

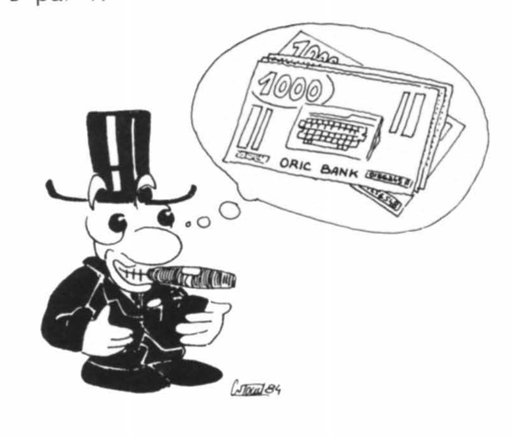
ligne 100
CALL # E6CA devient CALL # E76A

ligne 1080
CALL # E8Ø4 devient CALL # E93D

ligne 8120
SC$ = MID$(SC$,2) est inutile

lignes 12110 et 14750
Dans l'instruction PLOT remplacer le premier Ø par 1.

....




```
10 REM      $$$$$$$$$$$$$$$$$$$
12 REM      $$$ DOLLAR MAN $$$
15 REM      $$$$$$$$$$$$$$$$$$$
20 REM
30 REM *** CHRISTOPHE ANDREANI ***
40 REM
50 PRINTCHR$(6)CHR$(17)CHR$(20)
55 PRINTCHR$(19)
60 CLEAR:BI=2:NI=1
65 PRINTCHR$(19)
70 GOSUB12000
100 CALL#E6CA
150 LORES0
160 PRINTCHR$(19)
200 POKE46864,12:POKE46865,12:POKE46866,63:POKE46867,45:POKE46868,45
210 POKE46869,30:POKE46870,18:POKE46871,51
220 POKE46384,33:POKE46385,63:POKE46386,33:POKE46387,63:POKE46388,33
230 POKE46389,63:POKE46390,33:POKE46391,63
500 FOREE=1TO7:READD:FORFF=9TO29:PLOTFF,D,20:NEXTFF,EE
520 FORFF=3TO24:PLOT9,FF,20:PLOT29,FF,20:NEXT
530 PO=1:PP=0:OO=0:E=0:F=0
535 PLOT31,0,8
540 BO=3010+BB:TT=0
545 RESTORE
550 FOREE=1TO7:READD:NEXT
```

```
590 FORFF=1TO40:READXX,YY:PLOTXX,YY,38:NEXT
600 DATA3,7,11,15,19,23,24
605 DATA18,6,22,6,16,10,18,14,23,14,12,18,20,18,26,18
610 DATA18,7,22,7,18,8,22,8,18,9,22,9,18,10,22,10,16,11,16,12,16,13,16,14
620 DATA18,15,23,15,18,16,23,16,18,17,23,17,18,18,23,18,12,19,20,19,26,19
630 DATA12,20,20,20,26,20,12,21,20,21,26,21,12,22,20,22,26,22
640 PLOT11,6,36:PLOT27,6,36:PLOT11,10,36:PLOT27,10,36:PLOT11,14,36:PLOT27,14,36
650 PLOT15,18,36:PLOT27,18,36:PLOT14,22,36:PLOT24,22,36
690 PLOT2,25,16:PLOT4,25,16
700 ONBIGOTO720,710
705 GOTO790
710 PLOT4,25,98
720 PLOT2,25,98
780 PLOT5,26,5
790 PLOT15,26,"NIVEAU:"
850 NI$=STR$(NI)
855 NI$=MID$(NI$,2)
860 PLOT24,26,NI$
900 PLOT31,13,"$ =     P"
910 PLOT32,13,1
1000 X=16:Y=22:PO=1:CC=1:PP=0
1010 PLOT0,0,5
1050 PLOT1,0,"SCORE:"
1060 PLOT25,0,"BONUS:"
1080 CALL#E804
1100 A$=KEY$
1200 IFA$="q"THENE=-1:F=0
1300 IFA$="e"THENE=+1:F=0
1400 IFA$="p"THENF=-1:E=0
1500 IFA$="1"THENF=+1:E=0
1510 PLOT34,13,MID$(STR$(NI*100),2)
1550 BO=BO-10:IFINT(BO/100)=BO/100THENGOSUB16000
1560 IFBO<=0THEN10050
1570 IFBO=500THENPLOT31,0,12
1600 IFSCRN(X,Y+F)=16ORSCRN(X+E,Y+1)=16ORSCRN(X+E,Y+F)=20THENEF=0:GOTO5000
1700 EF=1
1900 IFSCRN(X+E,Y+F)=36THENSC=SC+(NI*100):PING:S(1)=S(1)+.15:CC=0:TT=TT+100
2000 IFSCRN(X+E,Y+F)=38THENPP=1
2100 SC$=STR$(SC)
2110 SC$=MID$(SC$,2)
2150 PLOT9,0,SC$
2900 PLOTX,Y,16
3000 X=X+E:Y=Y+F
3050 IFPP=2THENPLOTX-E,Y-F,38:PP=0
3100 IFPP=1THENPLOTX,Y-F,38:PP=2
5000 PLOTX,Y,98
5100 IFINT(TT/1000)=TT/1000ANDCC=0THEN8000
5200 IFEF<>0THENSHOOT:PLAY1,1,6,2000
6000 IFPO=1ANDRND(1)*3>2ANDRND(1)*20<1+S(1)THENSO=INT(RND(1)*(20-S(1))+5):PO=2
6100 IFPO=2THENOO=OO+1:IFOO>SOTHENPO=3
6150 IFPO=3THENOO=OO-2:IFOO<0THENPO=1:OO=0
6200 IFY=6ANDPO=3ORY=10ANDPO=3ORY=14ANDPO=3ORY=18ANDPO=3ORY=22ANDPO=3THEN10000
6300 ONPOGOTO6400,6500,6600
6400 PLOT19,2,18:GOTO1100
6500 PLOT19,2,19:GOTO1100
6600 PLOT19,2,17:GOTO1100
7000 GOTO1100
8000 FORFF=0TO9:READD:MUSIC1,4,D,13:WAIT10:NEXT:WAIT20:PLAY0,0,0,0
8050 DATA10,8,6,8,6,5,6,5,3,10
8100 CC=1:NI=NI+1:PO=1:OO=0
8110 PLOTX,Y,16
8120 SC=SC+K:SC$=STR$(SC):SC$=MID$(SC$,2):PLOT9,0,SC$
8150 WAIT200:BB=BB+500
8200 RESTORE:AA$=KEY$:GOTO530
10000 PLOT19,2,17:OO=0
```

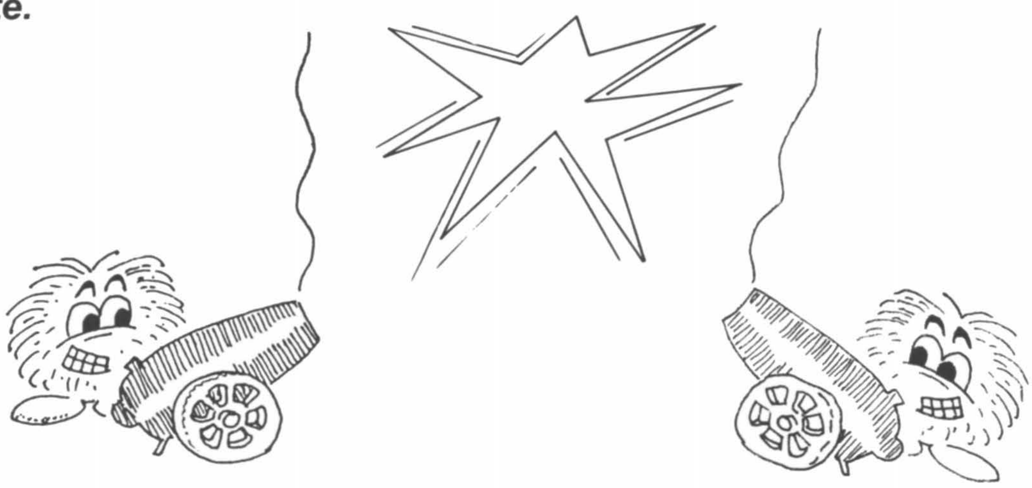
```
10050 PLAY1,0,0,0: FORFF=1TO3:MUSIC1,0,1,15:WAIT50:MUSIC1,0,1,0:WAIT9:NEXT
10052 BI=BI-1
10055 IFBI<>-1THENAA$=KEY$:RESTORE:S(1)=.15*10*NI:PO=1:PLOTX,Y,16:GOTO530
10100 PLOT9,2,"VOULEZ-VOUS REJOUER ?"
10110 II=0
10150 T$=KEY$
10160 II=II+1:IFII=8THENII=1
10170 PLOT5,2,II
10180 WAIT5
10200 IFT$="o"THENLORES0:RESTORE:AA$=KEY$:GOTO60
10300 IFT$="n"THEN END
10400 GOTO10150
12000 REM
12100 CLS:PAPER0:INK6
12110 PLOT0,0,3:PLOT0,10,4:PLOT0,12,5:PLOT0,24,1
12200 PRINT"          O R I C  1   presente:"
12300 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
12400 PRINTCHR$(4)CHR$(27);"N         D O L L A R   M A N"
12600 PRINTCHR$(4)
13100 PRINT:PRINT"            de C.Andreani"
13140 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
13150 PRINT:PRINT"Voulez-vous la liste des instructions?"
13200 GETT$
13300 IFT$="n"THENRETURN
13350 IFT$="o"THEN14000ELSE13200
14000 CLS:INK5:PRINT:PRINT"    VOTRE MISSION EST DE RECUPERER"
14010 PRINT"LE PLUS DE DOLLARS ($) POSSIBLE"
14500 PRINT:PRINT:PRINT:PRINT"Les commandes sont:"
14510 PRINT:PRINT"Q    =    ALLER A GAUCHE"
14520 PRINT:PRINT"E    =    ALLER A DROITE"
14530 PRINT:PRINT"P    =    MONTER"
14540 PRINT:PRINT"L    =    DESCENDRE"
14600 PRINT:PRINT:PRINT:PRINT"    ATTENTION!   VOUS NE DEVEZ PAS"
14610 PRINT"MARCHER SUR LES PLATEFORMES LORSQUE"
14620 PRINT"LE FEU EST ROUGE..."
14700 WAIT500:PRINT:PRINT:PRINT"APPUYEZ SUR UNE TOUCHE":GETT$
14710 INK6
14750 CLS:PLOT0,6,2:PLOT0,8,3:PLOT0,11,1
14800 PRINT:PRINT:PRINT"   LE FEU QUI SE TROUVE EN HAUT DE"
14850 PRINT"L'ECHAFAUDAGE PREND RESPECTIVEMENT"
14900 PRINT"LES COULEURS SUIVANTES:"
14950 PRINT:PRINT"VERT : VOUS POUVEZ ALLER PARTOUT
14960 PLOT7,6,6
15000 PRINT:PRINT"JAUNE: ATTENTION! LE FEU SERA BIENTOT"
15010 PLOT7,8,6
15050 PRINT"ROUGE"
15100 PRINT:PRINT"ROUGE: SI VOUS ETES SUR UNE DES"
15110 PLOT7,11,6
15150 PRINT"PLATEFORMES,VOUS ETES ELECTROCUTE!"
15200 PRINT:PRINT:PRINT"A CHAQUE TABLEAU,IL VOUS FAUDRA VOUS"
15250 PRINT"EMPARER DE TOUS LES $ AVANT QUE LE"
15300 PRINT"BONUS NE SOIT EGAL A ZERO.DANS LE CAS"
15350 PRINT"CONTRAIRE,VOUS SERIEZ LA AUSSI DETRUIT"
15400 PRINT:PRINT"   POUR VOUS ARRETER SUR UNE ECHELLE"
15450 PRINT"APPUYEZ SUR  LA TOUCHE  Q  OU SUR  E"
15460 WAIT500
15500 PRINT:PRINT" POUR COMMENCER,APPUYEZ SUR UNE TOUCHE":GETT$:RETURN
16000 BO$=STR$(BO):BO$=MID$(BO$,2):PLOT32,0,"      ":PLOT33,0,BO$:K=BO
16050 RETURN
```



FORT ORIC

par Gérald AUGUSTONI

Voici une variante du jeu "MUR DE BRIQUES". On attaque avec quatre raquettes depuis les quatre points cardinaux un bloc de briques diversement colorées : le FORT ORIC. Bastion difficile à détruire. Les couches successives rapportent 2, 4, 6, 8... points par brique atteinte et détruite.



A chaque changement de couche un bonus est calculé mais l'espace à défendre avec chaque raquette s'élargit en contrepartie. On actionne les raquettes avec deux flèches droite gauche ou haut bas.

Ce programme mérite d'être passé en langage machine et bon nombre d'amateurs sauront le faire. Tel qu'il est, en BASIC, avec un emploi considérable de PEEK et de POKE il permet de jouer sans trop de difficulté. La balle se déplace obliquement, vous remarquerez les effets d'écrasement.

Un son, modifiable en changeant Octave et Note avant les GOSUBs 6ØØØ rythme les chocs.

Affichage du nom du joueur, de son score, de la balle en jeu. Mémorisation et affichage du nom et du score du meilleur joueur. Le tout utilise la page TEXT à partir de caractères reconfigurés.

On a utilisé PRINT AT pour certains affichages. Les possesseurs d'ORIC-1 ayant encore la ROM V1.Ø utiliseront des PRINT, PLOT ou POKE en se servant des indications parues dans le n° 2 de MICR'ORIC en particulier. La difficulté provient du curseur qu'il faut rendre invisible. Il est donc dans le terrain au coin supérieur gauche.

Passons en revue le programme

lignes 0-4
PS puissance sonore
CH chance offerte (3 chances)

ligne 5
On initialise une zone mémoire qui servira à gérer le jeu.

lignes 10 à 99
Donnée pour reconfigurer les caractères.

lignes 100-110
Reconfiguration.

ligne 120
Envoi au S/P 7ØØØ, enregistrement du nom du joueur.

lignes 200-204
Préparation de la ligne supérieure de l'écran.

lignes 205-208
Initialisations diverses.

lignes 209-237
Dessin des contours

lignes 239-244
Affichage des raquettes.

lignes 250-265
Affichage du "FORT".

lignes 900-910
Réinitialisations.

lignes 100-1004
Initialisations aux changements de balle.

ligne 1005
Routine principale.

lignes 1010-1070
PF est la position de la balle.
FU le code du "caractère" rencontré.
H et V gèrent les déplacements horizontaux et verticaux.

lignes 1100-1130
Dessins de la balle diversement disposée.

ligne 1140
Effacement de la position précédente.

lignes 1150-2370
Déplacements divers.

lignes 3000-3060
Gestion du score : SC.
NB nombre de briques détruites.

lignes 3100-3130
Élargissement de la zone à défendre avec les raquettes.

lignes 5000 Divers sous programmes.
6000
7000
8000

```
0 CLS:PRINT@5,12;"";:INPUT"NIVEAU SONORE 1...15 ";PS
1 POKE618,10:CLS:PRINT@5,12;"UN PEU DE PATIENCE..."
4 INK3:PAPER0:CH=1
5 FORX=40000TO41120:POKEX,100:NEXT
10 DATAa,07,07,07,07,07,00,00,00
11 DATA^,00,01,03,03,03,01,00,00
12 DATA*,00,70,F8,F8,F8,70,00,00
15 DATAi,00,00,00,07,07,07,07,07
20 DATAb,38,38,38,38,38,00,00,00
25 DATAj,00,00,00,38,38,38,38,38
30 DATAc,00,00,00,00,3C,3C,3C,3C
35 DATAk,00,00,00,00,0F,0F,0F,0F
40 DATAd,3C,3C,3C,3C,00,00,00,00
45 DATAl,0F,0F,0F,0F,00,00,00,00
50 DATAe,3F,3F,3F,3F,3F,00,00,00
51 DATAv,00,00,00,00,38,3C,3C,3C
52 DATAw,00,00,00,00,07,0F,0F,0F
53 DATAx,0F,0F,0F,07,00,00,00,00
54 DATAy,3C,3C,3C,38,00,00,00,00
60 DATAf,0F,0F,0F,0F,0F,0F,0F,0F
70 DATAg,3C,3C,3C,3C,3C,3C,3C,3C
80 DATAh,00,00,00,3F,3F,3F,3F,3F
85 DATAm,00,00,00,0F,0F,0F,0F,0F
90 DATAn,3C,3C,3C,3C,3C,00,00,00
91 DATA@,1F,0E,00,00,00,00,00,00
92 DATAo,0F,0F,0F,0F,0F,00,00,00
93 DATAr,1E,3F,3F,1E,00,00,00,00
94 DATAu,00,00,00,00,1E,3F,3F,1E
95 DATAp,00,00,00,3C,3C,3C,3C,3C
96 DATAs,00,18,3C,3C,3C,3C,18,00
97 DATAq,00,0E,1F,1F,1F,0E,00,00
98 DATAt,00,06,0F,0F,0F,0F,06,00
99 DATA%,00,00,00,00,00,0E,1F,1F
```

N'hésitez pas à envoyer vos propositions de logiciels à :

MICRO-PROGRAMMES 5
82-84, bd des Batignolles
75017 PARIS

qui en retour vous fera une proposition de contrat si une publication est retenue.

AVIS AUX CRÉATEURS...

```
100 FORX=1TO29:READA$:Y=ASC(A$)
110 FORI=0TO7:READA$:A=VAL("#"+A$):POKE46080+I+8*Y,A:NEXTI,X
120 GOSUB7000
200 A$="RECORD           FORT ORIC                 "
201 FORX=1TO39:A=ASC(MID$(A$,X,1)):POKEX+48001,A:NEXT
202 CLS:POKE48001,0:POKE48009,4:POKE48015,2
203 FORX=2TO27:DOKE40*X+47999,771:DOKE39999+40*X,771:NEXT
204 POKE48000,17:POKE48040,3
205 FORX=0TO39:POKE40000+X,3:POKE41080+X,3:NEXT
206 N=0:C1=0:C2=0:C3=0:C4=0:C5=0:C6=0:SC=0
208 POKE49081,6
209 PLOT3,26,"SCORE:":PLOT11,26,1:PRINT@2,1;"";
210 PLOT1,0,"mhhhhhhhhhhhhhhh        hhhhhhhhhhhhhhhp"
220 A$="f                                      g "
225 FORX=1TO9:PLOT1,X,A$:NEXT
230 FORX=1TO9:PLOT1,X+15,A$:NEXT
235 PLOT1,25,"oeeeeeeeeeeeeeee        eeeeeeeeeeeeeeen"
237 PLOT16,26,2:PLOT17,26,NM$
239 FORP=48268TO48868STEP40:POKEP,16:POKEP-8000,2:NEXT
240 FORX=48041TO49041STEP40:IFPEEK(X)>90THENPOKEX,PEEK(X)+128
241 NEXT:FORX=48078TO49078STEP40:IFPEEK(X)>90THENPOKEX,PEEK(X)+128
242 NEXT:FORX=48042TO48077:IFPEEK(X)>90THENPOKEX,PEEK(X)+128
243 NEXT:FORX=49042TO49077:IFPEEK(X)>90THENPOKEX,PEEK(X)+128
244 NEXT
245 FORX=49058TO49060:POKEX,ASC("e"):NEXT
246 FORX=48058TO48060:POKEX,ASC("h"):NEXT
247 FORX=48481TO48561STEP40:POKEX,ASC("f"):NEXT
248 FORX=48518TO48598STEP40:POKEX,ASC("g"):NEXT
250 FORX=1TO7:FORI=0TO16-2*X
255 POKE48211+I+41*X,X+16: POKE40211+I+41*X,0
256 POKE48892+I-39*X,X+16: POKE40892+I-39*X,0
257 POKE48251+I*40+41*X,16+X: POKE40251+I*40+41*X,1
258 POKE48228+I*40+39*X,16+X: POKE40228+I*40+39*X,1
260 NEXT:NEXT
265 POKE48539,17:POKE48540,17:POKE48580,17:POKE48579,17
900 V1=48598:V2=48561:H2=48060:H1=49060
910 GG=1:FF=1:TR=3
1000 PP=INT(RND(3)*6):PO=#BD63+PP*41:POKEPO,ASC("q")
1003 V=-1:H=+1:AD=8000:SO=32:     TF=INT(RND(6)*2+1):IFTF=2THENV=-V
1004 PLOT25,26,5:PLOT26,26,"BALLE:  ":PLOT33,26,STR$(CH)
1005 GOSUB2100:GOSUB1010:GOSUB2100:GOTO1005
1010 PF=PO+H+V*40
1020 FU=PEEK(PF)
1022 IFFU>127THENFU=FU-128
1025 IFPEEK(PF-AD)=3THENPOP:GOTO5000
1030 IFFU=32THENDES=32:GOTO1500
1031 IFFU=109THENH=-H:V=-V:GOTO1150
1032 IFFU=112THENH=-H:V=-V:GOTO1160
1033 IFFU=111THENH=-H:V=-V:GOTO1170
1034 IFFU=110THENH=-H:V=-V:GOTO1180
1035 IFFU=16THENDES=16:GOTO1600
1040 IFFU=ASC("h")THENV=-V:GOTO1110
1041 IFFU=ASC("e")THENV=-V:GOTO1100
1045 IF FU=99 OR FU=107 THEN V=-1:H=-H:O=5:N=8:GOSUB6000:RETURN
1046 IF FU=97 OR FU=105 THEN H=-1:V=-V:O=5:N=8:GOSUB6000:RETURN
1047 IF FU=98 OR FU=106 THEN H=1:V=-V:O=5:N=8:GOSUB6000:RETURN
```

```
1048 IF FU=108 OR FU=100 THEN V=1:H=-H:O=5:N=8:GOSUB6000:RETURN
1050 IFFU=ASC("g")THENH=-H:GOTO1120
1051 IFFU=ASC("f")THENH=-H:GOTO1130
1060 IFPEEK(PF-AD)=0THENIFF=0THENV=-V:X=H:O=3:N=4:GOSUB6000
1070 IFPEEK(PF-AD)=1THENIFF=0THENH=-H:X=V*40:O=3:N=4:GOSUB6000
1074 GOSUB3000
1075 GOTO2000
1100 PA=PO+H:POKEPA,ASC("u"):GOTO1140
1110 PA=PO+H:POKEPA,ASC("r"):GOTO1140
1120 PA=PO+V*40:POKEPA,ASC("t"):GOTO1140
1130 PA=PO+V*40:POKEPA,ASC("s"):GOTO1140
1140 POKEPO,32:O=3:N=5:GOSUB6000:RETURN
1150 POKEPO,ASC("y"):O=5:N=8:GOSUB6000:RETURN
1160 POKEPO,ASC("x"):O=5:N=8:GOSUB6000:RETURN
1170 POKEPO,ASC("v"):O=5:N=8:GOSUB6000:RETURN
1180 POKEPO,ASC("w"):O=5:N=8:GOSUB6000:RETURN
1500 POKEPO,SO:POKEPF,ASC("q"):PO=PF:SO=DES:F=0:POKEPA,32:RETURN
1600 IFPEEK(PF-AD)=2THENPOKEPF+1,16:GOTO1500ELSE1500
2000 IFF=0THENPOKEPF,16:F=-1:POKEPO,SO:PO=PO+X:SO=PEEK(PO):RETURN
2005 POKEPO,SO
2006 SO=PEEK(PF)
2010 PO=PF:RETURN
2100 A=PEEK(#208):IFA=188ORA=180THEN2200
2105 IFA=172 ORA=156THEN2300
2110 RETURN
2200 IFPEEK(H1+2)=ASC("q")THENRETURN
2201 IFPEEK(H2-2)=ASC("q")THENRETURN
2202 IFPEEK(V1+40)=ASC("q")THENRETURN
2203 IFPEEK(V2-40)=ASC("q")THENRETURN
2210 IFPEEK(H1+1)=ASC("e")+128THENRETURN
2220 IFFF=1THEN2250
2225 POKEH1-2,97:POKEH1+1,ASC("b")
2230 POKEH2+1,106:POKEH2-2,ASC("i")
2235 POKEV1-80,99:POKEV1+40,ASC("d")
2240 POKEV2+40,108:POKEV2-80,ASC("k")
2245 FF=1:RETURN
2250 POKEH1-2,32:POKEH1+1,ASC("e")
2255 POKEH2+1,32:POKEH2-2,ASC("h")
2260 POKEV1-80,32:POKEV1+40,ASC("g")
2265 POKEV2+40,32:POKEV2-80,ASC("f")
2270 FF=0:H1=H1+1:H2=H2-1:V1=V1+40:V2=V2-40:RETURN
2300 IFFF=1THEN2350ELSEIFPEEK(H1-2)=ASC("q")THENRETURN
2301 IFPEEK(H2+2)=ASC("q")THENRETURN
2302 IFPEEK(V1-80)=ASC("q")THENRETURN
2303 IFPEEK(V2+80)=ASC("q")THENRETURN
2310 IFPEEK(H1-3)=ASC("e")+128THENRETURN
2322 H1=H1-1:H2=H2+1:V1=V1-40:V2=V2+40
2325 POKEH1+1,98:POKEH1-2,ASC("a")
2330 POKEH2-2,105:POKEH2+1,ASC("j")
2335 POKEV1+40,100:POKEV1-80,ASC("c")
2340 POKEV2-80,107:POKEV2+40,ASC("l")
2345 FF=1:RETURN
2350 POKEH1+1,32:POKEH1-2,ASC("e")
2355 POKEH2-2,32:POKEH2+1,ASC("h")
2360 POKEV1+40,32:POKEV1-80,ASC("g")
2365 POKEV2-80,32:POKEV2+40,ASC("f")
```

```
2370 FF=0:RETURN
3000 IFF<>0THENRETURN
3006 NB=NB+1:IFNB=256THENPOP:GOTO8000
3010 SC=SC+2*(FU-16): PLOT10,26,STR$(SC)
3015 IFFU=18THENC1=C1+1:IF C1=1THENSC=SC+N:GOTO3100
3017 IFFU=19THENC2=C2+1:IF C2=1THENSC=SC+2*C1:GOTO3100
3020 IFFU=20THENC3=C3+1:IF C3=1THENSC=SC+3*C2:GOTO3100
3030 IFFU=21THENC4=C4+1:IF C4=1THENSC=SC+4*C3:GOTO3100
3040 IFFU=22THENC5=C5+1:IF C5=1THENSC=SC+5*C4:GOTO3100
3050 IFFU=23THENC6=C6+1:IF C6=1THENSC=SC+6*C5:GOTO3100
3060 RETURN
3100 PING
3105 TR=TR+1:POKE48059+TR,32:POKE48060-TR,32
3110 POKE49059+TR,32:POKE49060-TR,32
3120 POKE48558+TR*40,32:POKE48598-TR*40,32
3130 POKE48521+TR*40,32:POKE48561-TR*40,32:RETURN
4000 OP=48500:I1=1:POKEOP,ASC("q")
5000 CH=CH+1:IFCH=4THENCH=1:GOTO5020
5010 ZAP:POKEPO,SO:WAIT200:GOTO1000
5020 IFSC>SMTHEN5030
5025 CLS:PRINT@5,7;NM$" A FAIT "SC" POINTS.":WAIT150:CLS:GOTO5050
5030 SM=SC:SM$=STR$(SM):L=LEN(SM$)
5035 FORI=1TO11:POKE48027+I,32:NEXT:POKE48026,3
5040 FORI=1TOL:POKE48008+I,ASC(MID$(SM$,I,1)):NEXT
5045 FORI=1TOLEN(NM$):POKE48027+I,ASC(MID$(NM$,I,1)):NEXT
5050 PRINT@12,12;"ON CONTINUE (O/N)":GETA$
5055 IFA$="O"THENCLS:GOTO5065
5060 IFA$<>"N"THENCLS:GOTO5050
5062 IFA$="N"THENEND
5065 PRINT@7,12;"NOUVEAU JOUEUR (O/N) ":GETA$:IFA$<>"N"THENGOSUB7000
5066 GOTO202
6000 MUSIC1,O,N,PS:PLAY1,0,0,0:WAIT12:PLAY0,0,0,0:RETURN
7000 CLS:PING:PRINT@5,12;"NOM DU JOUEUR :";:INPUTNM$
7010 IFLEN(NM$)>10THENNM$=LEFT$(NM$,10)
7020 RETURN
8000 CLS:PRINT@10,12;"FELICITATIONS!"
8010 PRINT:PRINTNM$" EST UN VERITABLE CHAMPION."
8020 O=1:N=1:FORI=1TO6:GOSUB6000:O=O+1:N=N+1:NEXT:GETA$
```

RASE MOTTES

par Christophe ROUX

Sur fond de ciel bleu nuit, votre bombardier blanc lumineux survole une agglomération dense. Vous devez détruire les buildings. A chaque passage vous perdez de l'altitude. Un seul projectile peut figurer sur l'écran à la fois. Si vous ne réussissez pas vous vous écrasez inéluctablement sur les vestiges de la ville. Vous avez trois avions, ce jeu se prête à la compétition. Divers niveaux de difficultés sont prévus.

Le listing fourni convient à l'ATMOS. Pour l'ORIC-1 il suffit de changer les lignes suivantes :



```
1030 POKE616,0:PRINT:POKE617,15:PRINTCHR$(8);CHR$(27);"QSCORE:";N;CHR$(27);"T";
CHR$(9)

2010 POKE616,13:PRINT:POKE617,8:PRINTCHR$(4);CHR$(27);"A"CHR$(27);"N IL VOUS RE
STE ";

2090 POKE616,7:PRINT:POKE617,13:PRINTCHR$(4);CHR$(27);"A"CHR$(27);"N GAME OVER"
;

2110 POKE616,7:PRINT:POKE617,13:PRINTCHR$(4);CHR$(27);"A";CHR$(27);"N BRAVO!";

2125 POKE616,10:PRINT:POKE617,2:PRINT

4000 POKE616,5:PRINT:POKE617,13:PRINTCHR$(4);CHR$(27);"A";CHR$(27);"J RECORDS:"
;

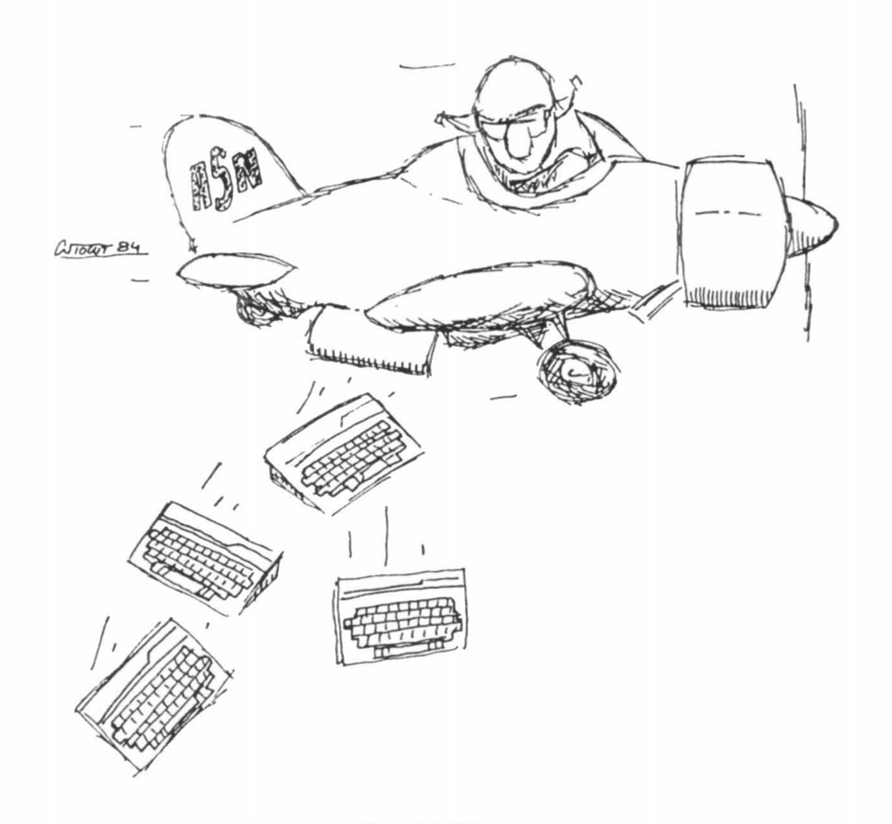
4020 POKE616,12:PRINT:POKE617,8:PRINT"NO 1:                                    "
4030 POKE616,15:PRINT:POKE617,8:PRINT"NO 2:                                    "
4040 POKE616,18:PRINT:POKE617,8:PRINT"NO 3:                                    "
4050 POKE616,21:PRINT:POKE617,8:PRINT"NO 4:                                    "

4500 POKE616,12:PRINT:POKE617,15:PRINTL,L$
4510 POKE616,15:PRINT:POKE617,15:PRINTK,K$
4520 POKE616,18:PRINT:POKE617,15:PRINTJ,J$

4530 POKE616,21:PRINT:POKE617,15:PRINTH,H$

4550 PRINT:PRINT"(1) DEPART                         (2) ARRET":INPUTN$
```

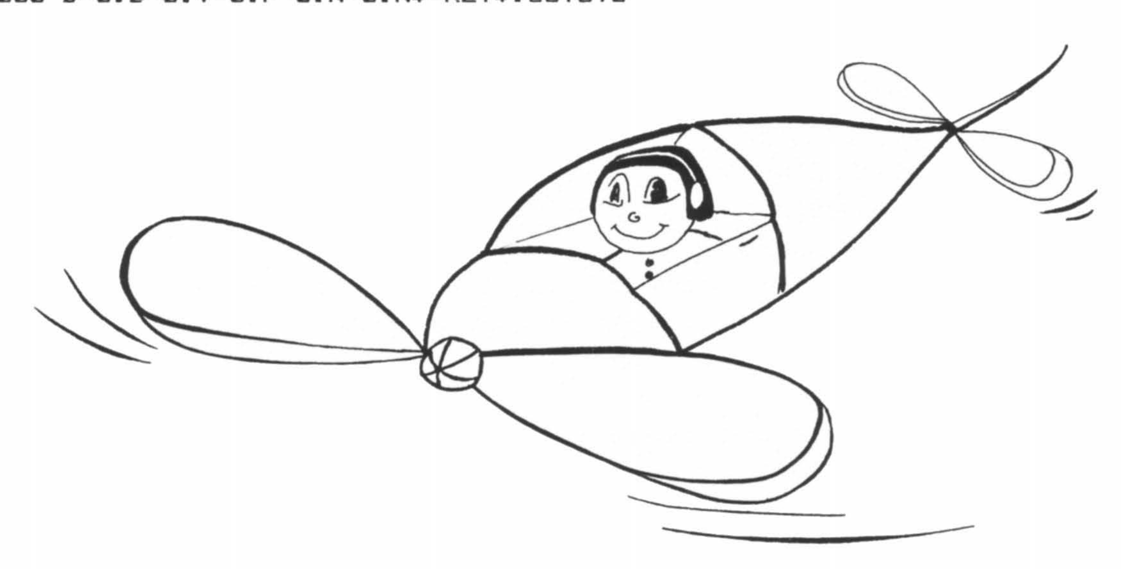
La seule différence est liée à l'emploi de PRINT AT sur l'ATMOS, instruction qui fait défaut sur ORIC-1 et que l'on simule en jouant sur le contenu des mémoires 616 et 617 comme vous pouvez le voir.




```
 1 L$="LE JEU DU RASE-MOTTES DE MICR'ORIC"
 2 POKE48000,22:POKE48001,1:POKE48002,12
 3 FOR X=3TO36:POKE48000+X,ASC(MID$(L$,X-2,1)):NEXT X
 5 POKE48037,32:POKE48038,32:POKE48039,32
 6 HIMEM#97FF
 7 TEXT:INK1:PAPER0
 10 PRINT CHR$(12):PRINT:PRINT:PRINT:PRINT
 20 PRINT CHR$(4);CHR$(27);CHR$(74);"              RASE MOTTE";CHR$(4)
 30 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
 40 PRINT "             AUTEUR : C.ROUX"
 50 FORX=0TO71:READ Y:POKE46856+X,Y:NEXT X
 55 N$=KEY$
 60 PRINTCHR$(12):PRINT:PRINT:PRINT:PRINT:PRINTCHR$(4);CHR$(27);"J
BUT DU JEU:";
 63 PRINT CHR$(4)
 65 PRINT:PRINT:PRINT"VOUS ETES AUX COMMANDES D'UN BOMBAR-  DIER QUI SURVOLE UNE
 VILLE E";
 67 PRINT"NNEMIE."
 70 PRINT"VOUS DEVEZ DEMOLIR LES BUILDINGS POUR ATTERRIR;POUR CELA LARGUEZ VOS B
OM";
 75 PRINT"BES EN APPUYANT SUR LA BARRE D'ESPACE.    ATTENTION,VOUS NE POUVEZ LAR
GU";
 76 PRINT"ER DEUX BOMBES EN MEME TEMPS."
 80 PRINT:PRINT:PRINT"APPUYEZ SUR UNE TOUCHE POUR JOUER":GET N$
 90 PRINTCHR$(12):INK0:PAPER4:POKE618,10:FOR Y=1TO9:X=INT(RND(1)*7)+5:REPEAT
 100 PLOT6+3*Y,26-X,"hi":X=X-1:UNTIL X=-1:NEXT
 110 FOR S=BTO26
 120 SOUND1,3910,10:SOUND2,3910,10:PLAY7,0,0,0
 130 FOR Z=1TO32
 140 PLOT Z,S," abcdef"
 150 IF Z=32 THENGOTO190
 160 IF SCRN(Z+7,S)=105THENGOTO2000
 170 IFS=26 THENGOTO190
 180 GOSUB600
 190 NEXT Z
```

```
200 PLOT32,S,"        "
210 NEXT S
300 GOTO2100
500 DATA12,14,15,15,15,15,63,63,0,0,0,32,56,63,63,63,0,0,0,0,0,63,63,63
510 DATA0,0,0,7,63,63,63,63,0,0,0,0,48,62,63,62,0,0,0,0,0,0,32,0
520 DATA31,31,14,14,14,14,14,4
530 DATA63,49,49,49,63,63,63,63,63,35,35,35,63,63,63,63
600 IF KEY$=" " THEN GOTO620
610 RETURN
620 T=S+1:R=INT((Z+1)/3)*3+3:X=FRE("")
630 IF Z=32 THEN GOTO640 ELSE GOTO650
640 Z=1:S=S+1
650 X=0:Y=0:FOR P=T TO26
660 PLOTR,P-1,"  "
670 PLOTR,P,"g"
680 IF P=26 THEN GOTO710
685 IF X=1 THEN GOTO700
690 IF SCRN(R,P+1)=104 THEN X=1 ELSEGOTO710
700 Y=Y+1:N=N+10:SHOOT:IF Y=4 THENGOTO780
710 PLOTZ,S," abcdef":Z=Z+1:IF Z<32THENGOTO730
720 PLOTZ-1,S,"         ":Z=1:S=S+1:GOTO740
730 IF SCRN(Z+7,S)=105 THEN GOTO2000
740 PLOTZ,S," abcdef":Z=Z+1:IF Z<32THENGOTO760
750 PLOTZ-1,S,"         ":Z=1:S=S+1:GOTO770
760 IF SCRN(Z+7,S)=105 THEN GOTO2000
770 NEXT P
775 P=P-1
780 PLOTR,P,"  "
790 Z=Z-1
1000 REM ******
1030 PRINT@15,1;"";:PRINTCHR$(8);CHR$(27);"QSCORE:";N;CHR$(27);"T";CHR$(9)
1045 N$=KEY$
1050 SOUND1,3910,10:SOUND2,3910,10:PLAY7,0,0,0
1060 RETURN
2000 EXPLODE:FOR X=0TO7:PAPERX:WAIT5:INKX:NEXTX:PAPER4:INK0
2001 IF V=2THENGOTO2007
2002 IF V=1THENGOTO2005
2003 IF N/1000>5THEND=D-1:V=1
2005 IF N/1000>10THEND=D-1:V=2
2007 D=D+1:IF D=3 THEN GOTO2090
2010 PRINT@5,11;"";:PRINTCHR$(4);CHR$(27);"A"CHR$(27);"N IL VOUS RESTE ";
2020 PRINT3-D" AVION(S)"
2030 GOTO3200
2090 PRINT@13,7;"";:PRINTCHR$(4);CHR$(27);"A"CHR$(27);"N GAME OVER";
2091 PRINTCHR$(4)
2092 POKE618,3
2094 WAIT500:GOTO3800
2100 LORES0:PRINT CHR$(27);"L":POKE618,10
2110 PRINT@13,8;""::PRINTCHR$(4);CHR$(27);"A";CHR$(27);"N BRAVO!";
2115 PRINT CHR$(27);"N";CHR$(4)
2120 IF F=1200 THEN F=1000
2125 PRINT@2,12;"";:PAPER2:INK4
2130 F=F+200:GOTO2000+F
2200 PRINT"MAINTENANT ECLATEZ-VOUS AVEC LE NIVEAU2"
2210 B=3:GOTO3200
2400 PRINT"MAINTENANT SCRATCHEZ-VOUS SUR LES      BUILDINGS DU NIVEAU 3"
2410 B=5:GOTO3200
2600 PRINT"VOUS NE DEVRIEZ-PAS VOUS ENTRAINER     AUTANT.PLACE AU NIVEAU 4"
2610 B=7:GOTO3200
2800 PRINT"SOIT VOUS ETES UN GENIE SOIT VOUS AVEZ UNE CHANCE INOUIE.NIVEAU 5"
2810 B=8:GOTO3200
3000 PRINT"VOUS JOUEZ TROP.VOUS ALLEZ EN DEVENIR MALADE.NIVEAU 6"
3100 B=10
```

```
3200 PLAY7,0,0,0:SOUND1,3910,15:SOUND2,3910,15:PLAY7,0,7,1:X=40
3280 X=X*1.09:SOUND1,X,13:SOUND2,(X-1),13:SOUND3,(X+10),13:IFX<480 GOTO3280
3290 WAIT100:SOUND1,0,0:SOUND2,0,0:SOUND3,0,0:PLAY0,0,0,0
3295 N$=KEY$:X=FRE("")
3300 GOTO90
3800 PRINT CHR$(12)
4000 PRINT@13,5;"";:PRINTCHR$(4);CHR$(27);"A";CHR$(27);"J RECORDS:";
4005 X=FRE("")
4010 PRINT CHR$(4)
4020 PRINT@8,13;"NO 1:                              "
4030 PRINT@8,16;"NO 2:                              "
4040 PRINT@8,19;"NO 3:                              "
4050 PRINT@8,22;"NO 4:                              "
4060 PRINT:INPUT"VOTRE NOM S'IL VOUS PLAIT";N$:IF LEN(N$)>15 THENGOTO4060
4065 IF N<=H THENGOTO5000
4070 IF N>K THEN GOTO4110
4080 IF N>J THENGOTO4090
4083 H=N:H$=N$:GOTO4500
4090 H=J:J=N:H$=J$:J$=N$
4100 GOTO4500
4110 IF N>L THENGOTO4140
4120 H=J:J=K:K=N:H$=J$:J$=K$:K$=N$
4130 GOTO4500
4140 IF N=H OR N=J OR N=K OR N=L THEN 4500
4145 H=J:J=K:K=L:L=N:H$=J$:J$=K$:K$=L$:L$=N$
4500 PRINT@15,13;L,L$
4510 PRINT@15,16;K,K$
4520 PRINT@15,19;J,J$
4525 A=FRE("")
4530 PRINT@15,22;H,H$
4532 R$="HI-SCORE:":Y$=STR$(L)+" "+L$
4533 FOR X=0TO39:POKE48000+X,32:NEXT X
4534 POKE48000,22:POKE48001,5:POKE48002,12
4537 FOR X=3TO11:POKE48000+X,ASC(MID$(R$,X-2,1)):NEXT X
4544 FOR X=0TO((LEN(Y$))-1):POKE48012+X,ASC(MID$(Y$,X+1,1)):NEXT X
4550 PRINT@7,1;"(1) DEPART                                   (2) ARRET":INPUTN$
4560 IF N$="1"THENGOTO5000
4570 IF N$="2"THEN GOTO4590
4580 GOTO4550
4590 FORX=48000TO48039
4600 POKEX,32:NEXT
4610 PRINT CHR$(12):END
5000 D=0:B=0:V=0:F=0:N=0:N$=KEY$:GOTO90
```



VARIABLES NUMÉRIQUES

VARIABLES NUMÉRIQUES PRINCIPALES :

S : Ordonnée du bombardier
Z : Abscisse du bombardier
R : Ordonnée de la bombe
P : Abscisse de la bombe
N : Score
B : Ordonnée du bombardier en début de tableau
L : 1er score
K : 2e score
J : 3e score
H : 4e score
A : Nombre d'otages détruits par une bombe
D : Nombre de bombardiers détruits
F : Niveau de jeu (adresse du sous-programme de niveau de jeu)
V : Bombardier en plus grand score = 5000 et 10000
T : Ordonnée de la bombe au largage

AUTRES VARIABLES :

X et Y : Petites boucles

VARIABLES CHAINES :

N$: Nom du joueur, aussi utilisée pour N$ = KEY$ (annule le largage de la bombe)
L$: Nom du meilleur joueur
K$: Nom du 2e meilleur joueur
J$: Nom du 3e meilleur joueur
H$: Nom du 4e meilleur joueur

CARACTÈRES RECONFIGURÉS :

a, b, c, d, e, f : Bombardier
h, i : Building
g : Bombe

STRUCTURE DU PROGRAMME :

1 à 40 : Titre
50 : Reconfiguration des caractères
55 à 80 : Affichage de la règle du jeu
90 à 100 : Dessin des buildings
110 à 300 : Déplacement du bombardier
500 à 530 : DATA des caractères reconfigurés
600 à 1060 : Déplacement du bombardier lorsqu'une bombe tombe
2000 à 2030 : Décompte des bombardiers restants
2090 à 2094 : Fin de partie
2100 à 3300 : Commentaire en fin de tableau, niveau de jeu
3800 à 4530 : Affichage des meilleurs scores
4532 à 4544 : Affichage de HI-SCORE
4550 à 5000 : Début de la partie suivante

MICR'ORIC a examiné pour vous quelques nouveautés

PLASTA BLASTA Arcadia

Micro-programmes 5

Les anglais assimilent l'Italie au Cosmos!

Ce jeu d'habileté ressemble à une guerre spatiale mais il se déroule dans un univers de sauce tomate! Vous dirigez un distributeur de sauce et vous pouvez asperger tout autour de vous, anéantissant les voleurs de boîtes (vos réserves) que sont de drôles de formes molles qui s'agittent en tous sens. Vos ennemis sont des dévoreurs. Votre réserve peut être régénérée en rejoignant des cibles particulières qui surgissent aléatoirement.

Ce jeu demande beaucoup d'habileté, il est bien sûr en couleurs, bruité assez agréablement, en langage machine. On règle la difficulté, le niveau sonore. On y trouve la page des records, et l'on dispose de 9 vies par parties. Ce jeu est annoncé comme le premier d'une trilogie.



LA MAISON DE LA MORT

Tansoft - Micro-programmes 5

Voici un jeu d'aventures de forme classique.

Vous avez à explorer une maison que l'on prétend "hantée". Vous entrez les commandes par des directions N, S, E, O ou par des verbes, le test se fait sur le groupe de 4 lettres au plus du début du mot. En langage BASIC, avec listing accessible, ce qui est bien commode en cas d'anomalies de chargement ce jeu vous propose des heures de recherches.

AUTEUR

Micro-programmes 5

Voilà un traitement de texte en langage machine très performant.

Dès la première heure d'utilisation vous serez à même d'utiliser bon nombre des possibilités offertes et qui sont classiques. Il vous faudra plus de temps pour découvrir les subtilités très intéressantes et assimiler les réglages de mise en page sur imprimante (interfacée CENTRONICS).

A tout moment vous connaissez l'espace disponible, le nombre de mots écrits. La recherche est très rapide, vous pouvez changer des mots, des lettres. L'effacement est sophistiqué : une lettre à la fois, ou un mot, ou une phrase ou un paragraphe ou tout le texte jusqu'à la fin, etc. L'insertion est très aisée. On peut même programmer les attributs d'affichage à l'écran (couleur, double hauteur, etc.) et ceux de l'imprimante. Pour ceux qui ont une imprimante possédant les caractères spéciaux au français, il est facile de reconfigurer les caractères (par exemple @ en à) avant de charger "AUTEUR" on a ainsi à l'écran et sur l'imprimante le même texte. Voilà un utilitaire fortement recommandable. Il est distribué avec un fascicule d'utilisation en français. La sauvegarde des textes se fait sur cassettes.

ESQUIVE

Une nouvelle version vous est proposée, adaptée à l'ORIC-1 ou à l'ATMOS.

Le jeu est plus ou moins complexe à la demande. En fait on vous propose plusieurs jeux en un.

Ce jeu de virtuosité, de tactique a des règles très simples mais pertinentes. Vous voilà au milieu d'une ville aux rues perpendiculaires sillonnées par des bolides qui vous laissent sur le ventre si vous ne les "esquivez" pas à temps. Au début, c'est facile, de nombreuses encoignures vous permettent de vous mettre à l'abri. Ce faisant vous collectionnez des clefs, des dollars, etc., le thème faisant de vous un glaneur expert. Pour aller d'un tableau à l'autre il vous faudra réussir même dans les coins, à vous de découvrir la méthode. Lorsqu'on réussit on peut être légitimement fier.

Écrit en un BASIC rigoureux avec juste ce qu'il faut de langage machine pour que votre virtuosité ne soit pas déçue ce jeu ne vous lassera pas de sitôt.

SCORBUTT

Micro-programmes 5

Un jeu à se rendre malade!

Votre organisme est envahi par d'étranges particules. Vous devez colmater les issues, les empêchant ainsi de pénétrer, selon les niveaux vous avez 2 ou 3 passages à boucher. Ces étranges "GLOMS" dévorent votre énergie, mais si vous êtes en bonne condition physique, vous pouvez vous en faire des alliées, elles régénèrent votre organisme. Un monstre indestructible tantôt inoffensif tantôt dangereux rôde incessamment et peut vous aider dans les cas désespérés alors que souvent son action est plutôt destructrice.

Avec de l'adresse, de la vivacité, de l'ingéniosité... d'un peu de chance aussi vous parviendrez à survivre et vous entrerez dans un corps régénéré à chaque victoire. Ce jeu comporte une partie en BASIC et une longue zone en langage machine. Les niveaux de vitesse de déplacement et de complexité sont réglables séparéments (de Ø à 9). Son réglable, tableau des scores.

Un jeu propice aux compétitions entre amis.

Les paramètres sont très nombreux, on n'a jamais l'impression de jouer la même partie. Les règles sont multiples, on les découvre progressivement.

Se joue à l'aide de 4 touches du clavier.

LE SCEPTRE D'ANUBIS

Micro-programmes 5

Un splendide jeu d'aventures magnifiquement illustré.

Vous êtes archéologue, vous voilà en Égypte à la recherche du tombeau d'un pharaon. Vous circulez dans les salles couvertes de hiéroglyphes et contenant divers trésors. Vous devez, à l'égal de Champollion déchiffrer ces hiéroglyphes.

Évidemment la malédiction des pharaons plane sur vous, profanateur de tombeau : des pièges mortels vous sont tendus, d'autres sont évitables.

Ce sont des heures patientes de recherche qui vous attendent et si vous vous perdez en route vous devez reprendre au tout début. Votre mémoire pourra vous être utile mais les circonstances de votre nouvelle aventure ne seront plus vraiement les mêmes. Votre ordinateur pourra rester allumé de longs jours si vous vous obstinez. En cas de réussite votre avenir est assuré, on vous propose un excellent emploi en rapport avec vos compétences...



le Sceptre d'Anubis

MICRO PROGRAMMES 5

LE SCEPTRE D'ANUBIS :

JEU D'AVENTURE EN LANGAGE MACHINE.
LE « MUST » DE VOTRE GAMME DE LOGICIELS.

NOTRE LISTE DE LOGICIELS (compatibles Oric 1-Atmos). ILS VOUS PASSIONNERONT.

LIBRAIRIE : ORIC ET SON MICRO-PROCESSEUR.

JEUX EN FRANÇAIS :

CANONNADE/ORIC POT
CASIN' ORIC
CONTRE-ATTAQUE
CROQUEUR
ESQUIVES
GALAXIE
MAISON DE LA MORT
SCORBUTT
LE SCEPTRE D'ANUBIS
ULTIMA ZONE (V.F.)

UTILITAIRES AVEC MANUEL EN FRANÇAIS :

AUTEUR : traitement de texte.
ORIC CALC : tableur électronique.
ORIC GEST : logiciel de gestion familiale.
STAR : gestion de fichiers, mailing.

Micro Programmes 5 82-84, Bd des Batignolles 75017 Paris.

Nous recherchons des distributeurs

Un NOVEX® c'est personnel.



Brancher, débrancher, rebrancher ... et puis, installer, démonter et recommencer ... pour votre système micro ordinateur personnel, le téléviseur familial n'est qu'une solution (bâtarde, d'ailleurs), mais pas LA solution.

Sautez le pas. Avec votre propre moniteur couleur NOVEX, vous mettez votre système à l'abri : moins de manipulations : moins de pannes. En plus, la qualité de définition du NOVEX optimise les qualités de votre unité centrale.

Si ORIC a sélectionné NOVEX, il y a des raisons !

Les NOVEX sont compatibles avec les principaux micro ordinateurs du monde.



NOVEX 12/800 vert ou ambré.

Ecran : 31 cm
Visualisation :
h : 210 mm x 1 = 154 mm
Raccordement DIN/CINCH
Fréquence de balayage :
horizontal 14 500 à 17 000 Hz
vertical 50 à 80 Hz
Tension : 220/240 V sous 50 Hz
Réponse vidéo : 20 MHz ± 3 dB
Connecteurs : jack RCA
Entrée vidéo jack RCA
sortie vidéo
Dimension : l : 300 mm
h : 275 mm p : 300 mm
Prix habituellement pratiqué :
1090 F T.T.C.

NOVEX couleur 1414 - CL

Ecran : 37 cm
Visualisation 90 pouces carrés
Raccordement DIN/DIN
Résolution horizontale
couleur : 300 lignes,
monochromatique : 350 lignes
Entrée signal vidéo : 1,0 V
Entrée R.V.B. niveau TTL
Gamme de fréquence :
horizontal : 15 650 à 16 250 Hz
vertical : 48 à 65 Hz
Entrée son pick-up
Entrée vidéo :
prise d'entrée pick-up pour
signal couleur PAL
Dimensions : l : 380 mm
h : 357 mm p : 370 mm
Prix habituellement pratiqué :
2800 F T.T.C.

**Liste des revendeurs sur demande à ASN Diffusion
Z.I. «La Haie Griselle»
94470 BOISSY-ST-LEGER**